Dynamic Dependence Analysis : Modeling and Inference of Changing Dependence Among Multiple Time-Series

by

Michael Richard Siracusa

Submitted to the Department of Electrical Engineering and Computer Science in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Electrical Engineering and Computer Science
at the Massachusetts Institute of Technology

June 2009

Signature of Author: _____

Department of Electrical Engineering and Computer Science
May 12, 2009

Certified by: _____

John W. Fisher III, Principal Research Scientist of EECS
Thesis Supervisor

Accepted by: _____

Terry P. Orlando, Professor of Electrical Engineering
Chair, Department Committee on Graduate Students

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
JUN 2009		**00-00-2009 to 00-00-2009**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Dynamic Dependence Analysis : Modeling and Inference of Changing Dependence Among Multiple Time-Series	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Massachusetts Institute of Technology,Department of Electrical Engineering and Computer Science,Cambridge,MA,02139**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

In this dissertation we investigate the problem of reasoning over evolving structures which describe the dependence among multiple, possibly vector-valued, time-series. Such problems arise naturally in variety of settings. Consider the problem of object interaction analysis. Given tracks of multiple moving objects one may wish to describe if and how these objects are interacting over time. Alternatively, consider a scenario in which one observes multiple video streams representing participants in a conversation. Given a single audio stream, one may wish to determine with which video stream the audio stream is associated as a means of indicating who is speaking at any point in time. Both of these problems can be cast as inference over dependence structures. In the absence of training data, such reasoning is challenging for several reasons. If one is solely interested in the structure of dependence as described by a graphical model, there is the question of how to account for unknown parameters. Additionally the set of possible structures is generally super-exponential in the number of time series. Furthermore, if one wishes to reason about structure which varies over time, the number of structural sequences grows exponentially with the length of time being analyzed. We present tractable methods for reasoning in such scenarios. We consider two approaches for reasoning over structure while treating the unknown parameters as nuisance variables. First, we develop a generalized likelihood approach in which point estimates of parameters are used in place of the unknown quantities. We explore this approach in scenarios in which one considers a small enumerated set of specified structures. Second we develop a Bayesian approach and present a conjugate prior on the parameters and structure of a model describing the dependence among time-series. This allows for Bayesian reasoning over structure while integrating over parameters. The modular nature of the prior we define allows one to reason over a super-exponential number of structures in exponential-time in general. Furthermore, by imposing simple local or global structural constraints we show that one can reduce the exponential-time complexity to polynomial-time complexity while still reasoning over a super-exponential number of candidate structures. We cast the problem of reasoning over temporally evolving structures as inference over a latent state sequence which indexes structure over time in a dynamic Bayesian network. This model allows one to utilize standard algorithms such as Expectation Maximization, Viterbi decoding, forward-backward messaging and Gibbs sampling in order to efficiently reasoning over an exponential number of structural sequences. We demonstrate the utility of our methodology on two tasks: audio-visual association and moving object interaction analysis. We achieve state-of-the-art performance on a standard audio-.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**Same as Report (SAR)**	**190**	

Dynamic Dependence Analysis : Modeling and Inference of Changing Dependence Among Multiple Time-Series

by Michael Richard Siracusa

Submitted to the Department of Electrical Engineering
and Computer Science on May 12, 2009
in Partial Fulfillment of the Requirements for the Degree
of Doctor of Philosophy in Electrical Engineering and Computer Science

Abstract

In this dissertation we investigate the problem of reasoning over evolving structures which describe the dependence among multiple, possibly vector-valued, time-series. Such problems arise naturally in variety of settings. Consider the problem of object interaction analysis. Given tracks of multiple moving objects one may wish to describe if and how these objects are interacting over time. Alternatively, consider a scenario in which one observes multiple video streams representing participants in a conversation. Given a single audio stream, one may wish to determine with which video stream the audio stream is associated as a means of indicating who is speaking at any point in time. Both of these problems can be cast as inference over dependence structures.

In the absence of training data, such reasoning is challenging for several reasons. If one is solely interested in the structure of dependence as described by a graphical model, there is the question of how to account for unknown parameters. Additionally, the set of possible structures is generally super-exponential in the number of time series. Furthermore, if one wishes to reason about structure which varies over time, the number of structural sequences grows exponentially with the length of time being analyzed.

We present tractable methods for reasoning in such scenarios. We consider two approaches for reasoning over structure while treating the unknown parameters as nuisance variables. First, we develop a generalized likelihood approach in which point estimates of parameters are used in place of the unknown quantities. We explore this approach in scenarios in which one considers a small enumerated set of specified structures. Second, we develop a Bayesian approach and present a conjugate prior on the parameters and structure of a model describing the dependence among time-series. This allows for Bayesian reasoning over structure while integrating over parameters. The modular nature of the prior we define allows one to reason over a super-exponential number of structures in exponential-time in general. Furthermore, by imposing simple local or

global structural constraints we show that one can reduce the exponential-time complexity to polynomial-time complexity while still reasoning over a super-exponential number of candidate structures.

We cast the problem of reasoning over temporally evolving structures as inference over a latent state sequence which indexes structure over time in a dynamic Bayesian network. This model allows one to utilize standard algorithms such as Expectation Maximization, Viterbi decoding, forward-backward messaging and Gibbs sampling in order to efficiently reasoning over an exponential number of structural sequences. We demonstrate the utility of our methodology on two tasks: audio-visual association and moving object interaction analysis. We achieve state-of-the-art performance on a standard audio-visual dataset and show how our model allows one to tractably make exact probabilistic statements about interactions among multiple moving objects.

Thesis Supervisor: John W. Fisher III

Title: Principal Research Scientist of Electrical Engineering and Computer Science

Acknowledgments

I would like to thank my advisor, John Fisher, for many years of guidance, encouragement and generous support. I met John during my first visit to MIT and find it difficult to imagine getting through graduate school without him. His dedication to research, his willingness to dive into the details and his humor not only made him a perfect advisor, but also a great friend.

I would also like to thank Alan Willsky and Bill Freeman for their insight and advice while serving on my thesis committee. I was welcomed into Alan's research group shortly after my Masters and have greatly appreciated our weekly grouplet discussions. It always amazes me how all those ideas and information is stored in one man's brain. Talking with Bill or even passing him in a hallway always brings a smile to my face and reminds me how fun and exciting good research is.

I would also like to thank Alex Ihler, Kinh Tieu, Kevin Wilson, Biswajit Bose, Wanmei Ou and Emily Fox for always being willing to help, listen, and discuss research ideas. Archana Venkataraman and Kevin Wilson get a special thank you for getting through early drafts of this dissertation and coming back with great comments and improvements.

I have been fortunate to have had the opportunity to interact with some the most intelligent and unique people I know while at the Computer Science and Artificial Intelligence Laboratory and in the Stochastic Systems Group. I have learned so much from them and have come away with so many great friendships. I could not adequately thank them all here.

Most importantly, I would like to thank my family. All my successes would have been empty and road blocks insurmountable without their patience, love and encouragement. I especially would like to thank my nephews Simon, Ethan, Alex and Adam and my niece Cate for reminding me of what is important. Finally, I'd like to thank my sweet Ana for giving me a reason to finish and look towards the future.

Different aspects of this thesis was supported by the Army Research office under the Heterogenous Sensor Networks MURI, the Air Force Office of Scientific Research under the Integrated Fusion, Performance Prediction, and Sensor Management for Automatic Target Exploitation MURI, the Air Force Research Laboratory under the ATR Center, and the Intelligence Advanced Research Projects Activity (formerly ARDA) under the Video Analysis and Content Extraction project

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

In this dissertation we investigate the problem of analyzing the statistical dependence among multiple time-series. The problem is cast in terms of inference over structure used by a probabilistic model describing the evolution of these time-series, such as an undirected graphical model or directed Bayesian Network. We are primarily concerned with the *structure* of dependence described by the presence or absence of edges in a graphical model and as such we treat model parameters as nuisance variables. Building upon the large body of work on structure learning and dynamic Bayesian networks we present a model which yields tractable inference of statistical dependence relationships which change over time. We demonstrate the utility of our method on audio-visual association and object-interaction analysis tasks.

1.1 Motivation

When designing and building intelligent systems it would be beneficial to give them the ability to combine multiple sources of sensory information for the purpose of interpreting and understanding the environment in which they operate. Fusion of multiple sources of disparate information enables systems to operate more robustly and naturally. One inspirational example of such a system is the human being. Our five basic sensing modalities of sight, hearing, touch, taste and smell are combined to provide a view of our environment that is richer than using any single sense in isolation. In addition to being inherently multi-modal, human perception takes advantages of multiple sources of information within a single modality. We have two eyes, two ears and multiple pathways for our sense of touch. Moving beyond the act of sensing, the human brain is highly effective at extracting, combining and interpreting higher-level information from these sources. For example, using our visual input we are able to track multiple moving

objects and combine this information with what we hear in order to help identify the source of a sound.

There are two main challenges when designing systems that fuse information from multiple time-series. The first challenge is choosing how to represent the incoming data. The high dimensional nature of each data source makes processing the raw sensory input computationally burdensome. Extracting features from each data source can alleviate this problem by eliminating information that is irrelevant or inherently noisy. For example, computation could be reduced if one were able to extract the location of multiple objects in a scene to analyze their behaviors rather than processing the entire visual scene as a whole. Even if dimensionality is low or features are extracted, the important question of determine which features are informative for the specific task remains. There has been considerable work on the task of general feature selection [41, 8, 53, 55] and on learning informative representations across multiple data sources [29, 84, 30]. This dissertation will assume the challenge of representation has been addressed by one of these existing approaches or by carefully hand picking features which are sufficiently informative for the application of interest.

The second challenge is that of integration. Once features are extracted the there is a question of how information obtained from them can be effectively combined. Simple strategies can be devised assuming the data sources are independent of each other. While computationally simple, such strategies neglect to take advantage of any shared information among the inputs. The opposite extreme is to consider all possible relationships among the inputs. This comes at the cost of a more complex model for integration. Thus, a key task for fusing information among multiple data sources is to identify the relationships among them. If one knew that the dependence among the data sources this knowledge could be exploited to perform efficient integration. This is the main focus of this dissertation: developing techniques for identifying the relationships among the multiple sources of information. In many tasks, identify these relationships is the main result one is interested in.

Consider a scene in which there are several individuals, each of whom may be speaking at any given moment. Assume a recording of the scene produces a single audio stream in addition to a wide angle video stream. For each individual in the scene a separate video stream representing a region of the input video can be extracted. Given this data over a long period of time, one useful task is to determine who, if anyone, is speaking at each point in time. Humans are very effective at this task. The solution to

this audio-visual association problem has wide applicability to tasks such as automatic meeting transcription, social interaction analysis, and control of human-computer dialog systems. The semantic label identifying who is speaking can be related to associating the audio stream with many, one or none of the video streams. For example, the identification that "Victoria is speaking" indicates that the video stream for Victoria is associated with the audio stream.

Next, consider a scene with multiple moving objects. For example, think about a basketball game in which the moving objects are the players on each team and the ball. Given measurements of the position of these objects over a long duration of time, a question one may ask is: Which, if any, of these objects are interacting at each point in time? The answer to this question is useful for a variety of applications including anomalous event detection and automatic scene summarization. In this specific example of a basketball game, understanding the interactions among players can help one identify a particular play, which team is on offense, and who is covering whom. This is a common and natural task regularly performed by humans. Heider and Simmel [45] note that when presented a cartoon of simple shapes moving on a screen, one will tend to describe their behavior with terms such as "chasing","following" or being independent of one another. These semantic labels describe the interaction between objects and allow humans to provide a compact description of what they are observing.

Underlying both of these problems is the task of identifying associations or interactions among the observed time-series. The words *association* and *interaction* both describe statistical dependence. When we say the audio stream is associated with video stream for Victoria, we are implying the two time-series share information and should not be modeled as being independent of one another. Similarly, when we claim two objects are interacting we are implying some statistical dependence between them. For example, a "following" interaction implies a causal dependence between the leader's current position and the follower's future position.

Note that with each of these dependence relationship there is the question of identifying the nature of that dependence. There are many ways in which time-series can be dependent on each other. Associated with a particular dependence structure are a set of parameters describing that dependence. For example, these parameters will describe the tone of Victoria's voice in the audio stream and exactly how motion in the video stream is associated with changes in the audio. In the object interaction analysis scenario the parameters describing the causal dependence for a "following" behavior

will characterize how fast the objects are moving and how closely the leader is being followed. Our primary interest is in identifying the presence or absence of dependence rather than accurately characterizing the nature of that dependence describe by these parameters.

■ 1.2 Objective

We develop a general framework for the task of *dynamic dependence analysis*. The primary input to a system performing dynamic dependence analysis is a finite set of discrete-time vector-valued time-series jointly sampled over a fixed period of time. The output of the system is a description of the statistical dependence among the time-series at each point in time. This description may be in the form of a point estimate of or distribution over dependence structure. The techniques presented in this dissertation have additional inputs such as:

- The maximum number of possible dependence relationships the system should consider modeling the time-series changing among over time.
- A specified class of static dependence models that will be used to described a fixed dependence among the time-series.
- A prior on the dependence structures of interest and an optional prior on parameters describing the nature of dependence.

■ 1.3 Our Approach

Past approaches to the problem of audio-visual association have either been based on basic measures of dependence over fixed windows of time [46, 84, 30, 68] or on incorporating training data to help classify correct speaker association [68, 72]. Similarly, past approaches for object interaction analysis have relied on measuring statistical dependence among small groups of objects assuming the interaction is not changing [88] or on trained classifiers to detect previously observed interacting behaviors [66, 49, 48, 69].

Building upon previous work we wish to address the issues that arise when performing dependence analysis on data in which the structure may change over time. Furthermore we wish to do so in the absence of large training datasets. Approaches that rely on general measures of statistical dependence are powerful in that they do not rely on any application specific knowledge and can be adapted easily to many other

domains. Previous work has primarily focused on the use of windowed approaches [68, 72]. That is, given a window of time they measure dependence assuming it is static. By sliding this window over time such approaches attempt to capture changing dependence. When using such an approach the window size used must be long enough to get an accurate measure of dependence. At the same time it must be short enough as to not violate the assumption of a static dependence relationship being active within the window. That is, there is a tradeoff between window size and how fast dependence structures can change. Approaches that use training data have the advantage of being specialized to the domain of interest and may be able to overcome some of these issues by only needing a few samples to identify a particular structure. However, domain specific training data is not always available.

Our core idea is that rather than treating the problem as a series of windowed tasks of measuring static dependence, we cast the problem of dynamic dependence analysis in terms of inference on a *dynamic dependence model*. A dynamic dependence model explicitly describes dependence structure among time-series and how this dependence evolves over time. Using this class of models allows one to incorporate knowledge over all time when making a local decision about dependence relationships. Additionally, it allows one to take advantage of repeated dependence relationships from different points in time. We demonstrate the advantage of this approach over windowed analysis both theoretically and empirically.

■ 1.4 Key Challenges and Contributions

This dissertation addresses a set of key challenges and makes the following contributions:

How does one map the associations in the audio-visual task and the interactions in the object interaction analysis task to a particular statistical dependence structure?

We introduce two general static dependence models: a static *factorization model* (FactM) and *temporal interaction model* (TIM). The FactM describes multiple time-series as sets of independent groups. That is, it explicitly models how the distribution on time-series factorizes. In the audio-visual association task, reasoning over associations between the audio and video streams is related to reasoning over factorizations.

The TIM allows one to provide more details about the dependence among time-series. The model takes the form of directed Bayesian network describing the causal relationships between the current value of a single time-series based on the finite set of past values of other associated time-series. We treat the problem of identify interactions as one of identifying the details of these causal relationships.

How many possible structures are there and is it tractable to consider all of them?

The number of possible dependence structures among N time-series is super-exponential in N, $O(N^N)$. For many applications, such as the audio-visual association task, there is a small number of specific structures one is interested in identifying and thus inference is generally tractable. However, in other applications, one may want to consider all structures or significantly large subset that grows with N. We show that a super-exponential number of structures can be reasoned over in exponential-time in general by exploiting the temporal causality assumptions in our TIM. We further show that by imposing some simple restrictions on the structures considered one can reason over a still super-exponential number of them in polynomial-time.

How does one incorporate prior knowledge about dependence?

In some applications we may have some prior knowledge about which dependence structures are more likely than others. When considering a tractable set of structures a simple discrete prior distribution can be placed over them. However, this approach becomes intractable as the number of structures considered increases. Building upon the type of priors used for general structure learning [62, 54, 33] we present a conjugate prior on structures for a TIM which allows for efficient posterior updates.

When analyzing a large number of time-series, trying to uncover a single "true structure" active at any point in time is often less desirable than a full characterization of uncertainty. Using this class of priors we obtain a distribution over structure rather than a point estimate. In addition, the exact posterior marginal distributions and expectations can be obtained tractably. This allows for a detailed characterization of uncertainty in the statistical dependence relationships among time-series.

How does one separate the task of identifying dependence structure from that of learning the associated parameters?

That is, an issue which arises when inferring dependence relationships is how to deal with unknown parameters which describe the nature of dependence. In this dissertation we will explore two approaches. We show how a maximum likelihood approach can be used to obtain a point estimate of the parameters from the data being analyzed. We show theoretically and empirically how this approach causes problems for windowed dependence analysis techniques while it can help our dynamic dependence approach. Alternatively, we present a conjugate prior over parameters of a TIM and show how to tractably perform exact Bayesian inference integrating over all possible parameters.

How does one model dependence that may change over time?

Building on the large body of work on dynamic Bayesian networks (DBNs) and hidden Markov models (HMMs) we introduce a dynamic dependence model which contains a latent state variable that indexes structure. In this dissertation we assume the number of possible structures is known a priori. That is, the number of possible latent state values is assumed to be known. A simple Markov dynamic is placed on this latent variable to model how structure evolves over time. Adopting such a model allows one to infer changes in structure via the use of standard forward-backward message passing algorithms. This allows one to reason over an exponential number of possible sequence of dependence relationships in linear time.

Additionally, we formulate both audio-visual association and object interaction analysis tasks as special cases of dynamic dependence analysis. We show how state-of-the-art performance on a standard dataset for audio-visual speaker association can be achieved with our method. We demonstrate the utility of our approach in analyzing the interactions among multiple moving players participating in a variety of games.

■ 1.5 Organization of this Dissertation

We begin in Chapter 2 with a review of statistical dependence, graphical models, conjugate priors and time-series models. This will give the background needed for the rest of the dissertation. Chapter 3 introduces our two static dependence models which

assumed a fixed dependence structure over all time. We discuss conjugate priors over structure and detail inference on such models in both a classical maximum likelihood and Bayesian framework. Chapter 4 extends these models to become dynamic dependence models by embedding them in hidden Markov model. We show theoretically how such models have a distinct advantage over windowed approaches for dynamic dependence analysis. Details are given for maximum likelihood inference using Expectation Maximization and Bayesian inference using a Gibbs sampling approach. Chapters 5 and 6 present experiments using our models in audio-visual association and objective interaction analysis tasks respectively. We conclude with a summary and discussion of future work in Chapter 7.

Chapter 2

Background

In the previous chapter we introduced the general problem of analyzing dependence relationships among multiple time-series. In this chapter we give a brief overview of the basic technical background the rest of the dissertation relies on. We use statistical models to describe time-series and the relationships among them. That is, we describe time-series in terms of random variables and their associated probability distributions. We assume the reader has a basic understanding of probability theory which can found in introductory chapter of a wide variety of standard textbooks (c.f. [5, 70]).

We begin by defining statistical dependence. This is followed by a review of graphical models and how they can be used to encode the structure of dependence relationships among a set of random variables. Next, in Section 2.3, we discuss time-series in terms of discrete-time stochastic processes and present the use of a simple Markov model for describing temporal dependence. In Section 2.4 we overview the standard parametric models which will be used in this dissertation along with a discussion of conjugacy, inference and learning. Lastly, we briefly present select topics from information theory and their relation to the problem of analyzing statistical dependence.

The intent of this chapter is to provide a brief overview of the selected material. We refer the reader to standard text books for a more rigorous treatment and proofs (c.f. [5, 70, 21, 73]).

■ 2.1 Statistical Dependence

We start by defining statistical dependence. Intuitively, two random variables are dependent if knowledge of one tells you something about the other. To formalize this abstract concept on can more concretely define statistical *independence*. Two random variables, $\mathbf{x}$ and $\mathbf{y}$, are statistical independent if and only if their joint distribution is a

product of their marginal distributions:

$$p(\mathbf{x}, \mathbf{y}) = p(\mathbf{x})\, p(\mathbf{y}). \tag{2.1}$$

The notation $\mathbf{x} \perp\!\!\!\perp \mathbf{y}$ is used to denote statistical independence. Using the fact that $p(\mathbf{x}, \mathbf{y}) = p(\mathbf{x})\, p(\mathbf{y}|\mathbf{x}) = p(\mathbf{y})\, p(\mathbf{x}|\mathbf{y})$ this requirement is equivalent to saying that the conditional distribution of one random variable given the other is not a function of what is being conditioned on:

$$p(\mathbf{x}|\mathbf{y}) = p(\mathbf{x}) \quad \text{and} \tag{2.2}$$

$$p(\mathbf{y}|\mathbf{x}) = p(\mathbf{y}). \tag{2.3}$$

This definitions can be easily extended to more than two random variables. The joint of distribution of a collection of independent random variables factorizes as product of their marginals and all conditional distributions are independent of the random variables conditioned on.

A simple example of statistical independence is when $\mathbf{x}$ and $\mathbf{y}$ are the results of coin tosses of two separate coins. Each coin may have its own bias / probability of heads versus tails, but if they are two physically separate coins tossed in isolation, knowledge of whether $\mathbf{x}$ is heads or tails tells you nothing about the outcome $\mathbf{y}$.

Statistical dependence is simply defined to be the absence of statistical independence. That is, two random variables are dependent if their joint distribution is not the product of their marginal distributions. Going back to a simple coin example, let $\mathbf{x}$ be the result of a fair coin toss such that $p(\mathbf{x} = \text{heads}) = p(\mathbf{x} = \text{tails}) = .5$. Given $\mathbf{x}$, imagine placing a small weight on a second fair coin so that it is biased to be more likely to land on the same side as $\mathbf{x}$. Let $\mathbf{y}$ be the result of flipping this biased coin. To be concrete, assume this bias favors $\mathbf{x}$ with probability .9, $p(\mathbf{y} = \text{heads}|\mathbf{x} = \text{heads}) = p(\mathbf{y} = \text{tails}|\mathbf{x} = \text{tails}) = .9$ and $p(\mathbf{y} = \text{tails}|\mathbf{x} = \text{heads}) = p(\mathbf{y} = \text{tails}|\mathbf{x} = \text{heads}) = .1$. Here, $\mathbf{x}$ and $\mathbf{y}$ are statistically dependent. Our description of the experiment clearly indicates how knowledge of $\mathbf{x}$ tells us something about $\mathbf{y}$. Using Bayes' rule it is easy to show that the reverse is also true and knowledge of $\mathbf{y}$ also tells us something about $\mathbf{x}$. Specifically, $p(\mathbf{x} = \text{heads}|\mathbf{y} = \text{heads}) = .9 * .5/(.9 * .5 + .1 * .5) = .9$ while $p(\mathbf{x} = \text{heads}|\mathbf{y} = \text{tails}) = .1$.

Another important concept is that of *conditional independence*. Two random variables $\mathbf{x}$ and $\mathbf{y}$ are conditionally independent given another random variable $\mathbf{z}$, $\mathbf{x} \perp\!\!\!\perp \mathbf{y} \mid \mathbf{z}$, if and only if:

$$p(\mathbf{x}, \mathbf{y}|\mathbf{z}) = p(\mathbf{x}|\mathbf{z})\, p(\mathbf{y}|\mathbf{z}) \tag{2.4}$$

Figure 2.1. *Dependence:* Undirected graphical models depicting independent and dependent distributions for two random variables.

or equivalently

$$p(\mathbf{x}|\mathbf{y}, \mathbf{z}) = p(\mathbf{x}|\mathbf{z}) \quad \text{and} \tag{2.5}$$

$$p(\mathbf{y}|\mathbf{x}, \mathbf{z}) = p(\mathbf{y}|\mathbf{z}) \tag{2.6}$$

It is important to note that conditional independence does not imply independence and vice versa. We will give some specific examples in the following section when discussing Bayesian networks.

■ 2.2 Graphical Models

In the previous section we discussed the link between statistical independence of random variables and the structure of their joint distribution. Probabilistic graphical models combine the syntax of graph theory with probabilistic semantics to describe the dependence structure among a set of random variables. Graphs are used to described the overall joint probability of the random variables in terms of local functions on subgraphs. This local decomposition allows for efficient reasoning over the random variables represented. More importantly for this dissertation, graphical models provide a convenient way to specify and understand conditional independence relationships in terms of graph topology. For example, consider the two undirected graphical models depicted in Figure 2.1. It is intuitive and simple to read dependence information from these graphs.

In the following section we review basic graph theory to establish some notation. We then discuss different classes of graphical models, each of which uses a different graphical representation to described conditional independence. Again, these sections provide a quick overview and we refer the reader to standard text books for a more detailed treatment [73].

2.2.1 Graphs

A graph $G = \{V, E\}$ is a set of vertices/nodes V and a collection of edges E. Edges $(i, j) \in E$ connect two different nodes $i, j \in V$. Edges can be either undirected or directed. In undirected graphs, the edge (i, j) is in E if and only if (j, i) is as well. A vertex j is a *neighbor* of vertex i if $(i, j) \in E$ and the set neighbors for vertex i is simply the collection of all such $j \in V$. A *path* is defined as a sequence of neighboring nodes. If there is a path between all vertices the graph is said to be *connected*. For undirected graphs a *clique* $C \subset V$ is a collection of vertices for which all pairs are have an edge between them. If the entire graph forms a clique the graph is said to be *complete*.

In a directed graph, directed edges (i, j) connect a *parent* node i to its *child* j. Throughout the dissertation we will use the notation $\bar{E}$ to represent a set of directed edges and will represent them as arrows in figures. For example, see Figure 2.2(b). The set of all parents $\mathbf{pa}\left(i, \bar{E}\right)$ for a particular node i is the set of all $j \in V$ such that $(i, j) \in \bar{E}$. For brevity we will often drop the $\bar{E}$ and refer to this set as $\mathbf{pa}\left(i\right)$.

A useful generalization of a graph is a hypergraph. A hypergraph $H = \{V, F\}$ is composed of vertices V and associated hyperedges F. Each hyperedge $f \in F$ is specified by any non-empty subset of V. That is, each hyperedge can connected one or more vertices. Two vertices are neighbors in this hypergraph if they belong to a common hyperedge. We represent a hypergraph as a graph with extra square nodes for each hyperedge. All vertices which belong to that hyperedge are connected to the associated square node. This makes the graph *bipartite* in that edges only occur between vertices and the square hyperedge nodes. See Figure 2.2(c) for an example.

2.2.2 Factor Graphs

Factor graphs are a class of graphical models which use a hypergraph H to specify the form of the joint probability distribution on a set of random variables [58]. Let $\mathbf{x}_V$ represent the set of all random variables $\{\mathbf{x}_i | i \in V\}$. Given a set of hyperedges F, a factor graph represents the joint distribution $p\left(\mathbf{x}_V\right)$ as

$$p\left(\mathbf{x}_V\right) = \frac{1}{Z} \prod_{f \in F} \psi_f(\mathbf{x}_f), \tag{2.7}$$

where each ψ_f is a non-negative function of their arguments and Z, often referred to as

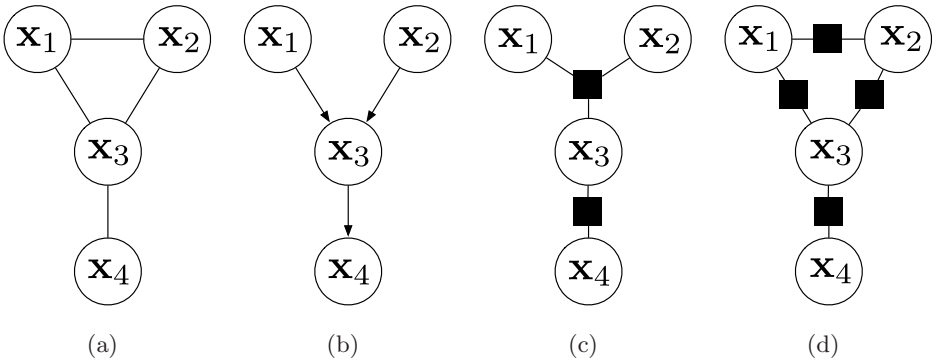


Figure 2.2. *Equivalent Graphical Models:* Three graphical models using different representations to describe the joint distribution of four random variables. (a) A markov random field (b) a directed Bayesian network. (c) a factor graph. (d) a second factor graph with the same neighbor relationships

the *partition function*, guarantees proper normalization. That is,

$$Z = \sum_{\mathbf{x}_V} \prod_{f \in F} \psi_f(\mathbf{x}_f) \tag{2.8}$$

Figure 2.2(c) shows an example in which

$$p\left(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4\right) = \frac{1}{Z} \psi_{1,2,3}(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3) \psi_{3,4}(\mathbf{x}_3, \mathbf{x}_4). \tag{2.9}$$

In the equation above, and in general, $\psi_f(\mathbf{x}_f)$ does not correspond to the marginal distribution $p\left(\mathbf{x}_f\right)$. However, in this dissertation we will use factor graphs in situations in which each random variable belongs to a single hyper edge and the ψ functions correspond to marginal distributions.

Conditional independence information can be read directly from a factor graph. Any two random variables $\mathbf{x}_i$ and $\mathbf{x}_j$ are conditional independent given a set $\mathbf{x}_Z$ for $Z \subset V$ if every path between $\mathbf{x}_i$ and $\mathbf{x}_j$ in the hypergraph passes through some vertex $k \in Z$. Two random variables are marginally independent if there is no path between them. In Figure 2.2(c), $\mathbf{x}_4 \perp\!\!\!\perp \mathbf{x}_1 \mid \mathbf{x}_3$. Using this fact it is simply to see that $\mathbf{x}_i$ is independent of all other random variables given its neighbors in the factor graph.

■ 2.2.3 Undirected Graphical Models

Undirected graphical models, generally referred to as Markov Random Fields (MRFs), encode conditional independence relationships in a graph $G = \{V, E\}$. Closely related to

factor graphs, MRFs represent the joint distribution as a product of potential functions on cliques c:

$$p\left(x_v\right) = \frac{1}{Z} \prod_c \phi_c(\mathbf{x}_c) \tag{2.10}$$

where, again, Z is the normalization constant. MRFs have the same conditional independence properties as factor graphs in that two random variables $\mathbf{x}_i$ and $\mathbf{x}_j$ are conditionally independent given a set of random variables that must be passed through for every path between $\mathbf{x}_i$ and $\mathbf{x}_j$.

Figure 2.2(a) shows an MRF with the same conditional dependence relationships as the factor graph in Figure 2.2(c). In fact, in general, there are many factor graphs which represent the same conditional independence relationships for a given MRF. Figure 2.2(d) shows a second factor graph which has the same dependence properties as Figure 2.2(a). Factor graphs have more flexibility in that they can explicitly specify how potentials on cliques factor, further constraining the space of possible joint distributions represented. This flexibility comes from their more general hypergraph specification.

■ 2.2.4 Bayesian Networks

Bayesian networks are graphical models which can encode causality via a directed representation. Given a directed graph $G = \{V, \bar{E}\}$ a Bayesian network represents a joint probability which factors as a product of conditional distributions:

$$p\left(\mathbf{x}_V\right) = \prod_{i \in V} p\left(\mathbf{x}_i | \mathbf{x}_{\mathbf{pa}(i)}\right). \tag{2.11}$$

If a node has no parents, $\mathbf{pa}\left(i\right) = \emptyset$, the we define $p\left(\mathbf{x}_i | \mathbf{x}_\emptyset\right) = p\left(\mathbf{x}_i\right)$. This factorization is only valid when $\bar{E}$ is *acyclic*. That is, it represents a graph in which there is no directed path from one node back to itself. Figure 2.2(b) depicts a Bayesian network with:

$$p\left(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4\right) = p\left(\mathbf{x}_1\right) p\left(\mathbf{x}_2\right) p\left(\mathbf{x}_3 | \mathbf{x}_1, \mathbf{x}_2\right) p\left(\mathbf{x}_4 | \mathbf{x}_3\right) \tag{2.12}$$

This decomposition exposes an underlying generative process.

Reading conditional independence from a Bayesian network is more complicated than for undirected models and factor graphs. The Bayes Ball algorithm provides a way of extracting independence relationships from Bayesian network structure [79]. One useful property is that a random variable is independent of all others conditioned

on its parents, its children, and its children's parents. A Bayesian network can be *moralized* into an MRF by connecting the parents of each node with an undirected edge and replacing all directed edges with undirected ones. Figure 2.2(a) is a moralized version of Figure 2.2(b).

Bayesian networks can explicitly represent certain statistical dependence relationships factor graphs and MRFs cannot. For example, consider the V structure found between $\mathbf{x}_1$,$\mathbf{x}_3$ and $\mathbf{x}_3$ in 2.2(b). A classic example with this dependence structure is a scenario when $\mathbf{x}_1$ and $\mathbf{x}_2$ represent the outcomes of two independent coin tosses. Let $\mathbf{x}_3$ be an indicator of whether $\mathbf{x}_1$ and $\mathbf{x}_2$ had the same outcome. Causally $\mathbf{x}_1$ and $\mathbf{x}_2$ are independent, $\mathbf{x}_1 \perp\!\!\!\perp \mathbf{x}_2$, but conditioning on knowing $\mathbf{x}_3$ they become dependent. Knowing whether the coin tosses where the same or not and the outcome of $\mathbf{x}_1$ tells you a lot about $\mathbf{x}_2$. In order to capture this relationship in an MRF, the three random variables must form a clique (adding an edge between $\mathbf{x}_1$ and $\mathbf{x}_2$). This is a case in which independence does not imply conditional independence. The reverse is true for $\mathbf{x}_1$ and $\mathbf{x}_4$ in Figure 2.2(b). They are statistical dependent but are conditionally independent given $\mathbf{x}_3$, $\mathbf{x}_1 \perp\!\!\!\perp \mathbf{x}_4 | \mathbf{x}_3$.

■ 2.3 Time-series

So far in this chapter we have been talking about statistical dependence among, and distributions for, general sets of random variables. In this dissertation we a interested in the dependence among time-series. A time-series can be modeled as a discrete time *stochastic process* whose value at time t is represented by a random variable, $\mathbf{x}_t$. A stochastic process is fully characterized by all joint distributions of the process at different points in time. A discrete stochastic process over a finite interval from 1 to T is completely specified in terms of its joint $p(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T)$. We will often use the notation $\mathbf{x}_{1:T}$ to denote such a sequence. If T is not fixed, this distribution would have to be specified for all possible T one expects to encounter. Such an approach is not tractable, and thus it is common to make certain assumptions about the temporal dependence.

One simple model for time-series is to consider each time point to be independent and identically distributed (i.i.d.) such that:

$$p(\mathbf{x}_{1:T}) = \prod_{t=1}^{T} p(\mathbf{x}_t) \tag{2.13}$$

While easy to represent, this approach does not capture the fact that information from

(a) First order (b) Second order

Figure 2.3. *Markov Models for Time-Series:* Directed Bayesian network (DBN) (a) represents a first order model and DBN (b) represents a second order model.

the past can help predict future time-series values. A *Markov model* is another tractable model which can capture some of this dependence. Consider the fact that the full joint distribution can always be factorized as

$$p\left(\mathbf{x}_{1:T}\right) = \prod_{t=1}^{T} p\left(\mathbf{x}_t | \mathbf{x}_{1:t-1}\right). \tag{2.14}$$

An r-th order Markov model is obtained if one assume the right-hand side is only dependent on the previous r time points. A first order model is simply

$$p\left(\mathbf{x}_{1:T}\right) = p\left(\mathbf{x}_1\right) \prod_{t=2}^{T} p\left(\mathbf{x}_t | \mathbf{x}_{t-1}\right). \tag{2.15}$$

and is depicted as a directed Bayesian network in Figure 2.3(a). A second order model is shown in Figure 2.3(b). Note that a zero order model is simply i.i.d.

We will use this class of models for our time-series throughout the dissertation. Note, here we are describing a single time-series $\mathbf{x}_{1:T}$. In Chapter 3 we will discuss models for jointly describing multiple time-series in which the temporal dependence is fixed but the dependence across time-series is unknown.

■ 2.4 Parameterization, Learning, Inference and Evidence

In the previous sections we discussed statistical dependence and how it can be encoded by a graphical model. We discussed how these models specify the form of the joint distribution in terms of local conditional probability distributions or potential functions. Implicitly each of these local distributions and potential functions has a set of parameters associated with it. The parameters fully specify its form. So far, we have been hiding these parameters in the notation.

Here will make this parameterization explicit and use the notation $p\left(\mathbf{x}|\Theta\right)$ rather than $p\left(\mathbf{x}\right)$. Similarly for conditional distributions we will use $p\left(\mathbf{x}|\mathbf{y}, \Theta_{\mathbf{x}|\mathbf{y}}\right)$ rather than

Figure 2.4. *Deterministic vs Random Parameters:* Two graphical models depicting two different treatments of parameters, Θ

(a) treats Θ as an unknown a deterministic quantity. (b) treats Θ as an unobserved random variable whose prior distribution is specified by known hyperparameters Υ.

$p(\mathbf{x}|\mathbf{y})$. For example, a more explicit representation for a directed Bayesian network on a set of random variables $\mathbf{x}_V$ is

$$p(\mathbf{x}_V|\Theta) = \prod_{i \in V} p\left(\mathbf{x}_i|\mathbf{x}_{\mathbf{pa}(i)}, \Theta_{i|\mathbf{pa}(i)}\right), \tag{2.16}$$

where Θ contains $\Theta_{i|\mathbf{pa}(i)}$ for all $i \in V$. Note that the actual parameterization is still implicit in this notation. Only the parameter values are explicitly denoted by Θ. For example, we must first say $p(\mathbf{x}|\Theta)$ is a Gaussian distribution, before one can identify that the parameters Θ describe the value of the mean and covariance for that distribution.

It is also important to point out that, while this notation explicitly specifies parameters with Θ, the structure described by $\bar{E}$ in Equation 2.16 is implicit. To be fully explicit one should use the notation $p\left(\mathbf{x}_V|\bar{E}, \Theta\right)$ instead. In future chapters of this dissertation we use this notation and focus on reasoning about this dependence structure. However, in this section, for simplicity, we will leave the structure implicit or consider it part of Θ.

Given a parameterization and parameter values one can calculate the probability of an observation of $\mathbf{x}$. However, throughout this dissertation we only be given observations of $\mathbf{x}$ without knowledge of the true underlying parameters. We can deal with this situation in one of three possible ways.

First, treating the parameters Θ as deterministic unknown quantities, we can attempt to *learn* them from the observed data. A common approach to learning is to maximize the likelihood of Θ. That is, if we denote $\mathcal{D} = \{\mathcal{D}_1, D_2, ...\mathcal{D}_N\}$ as N indepen-

dent observations of the random variable $\mathbf{x}$ we wish to find:

$$\hat{\Theta} = \arg\max_{\Theta} p\left(\mathcal{D}|\Theta\right) \tag{2.17}$$

$$= \arg\max_{\Theta} \prod_{n=1}^{N} p\left(\mathbf{x} = \mathcal{D}_n|\Theta\right) \tag{2.18}$$

We can also treat Θ as just another random variable. A prior, $p_0\left(\Theta|\Upsilon\right)$ can be placed on Θ to capture our prior belief on the value of Θ. Here, Υ is a set of *hyperparameters* used to specify this prior belief. This is depicted in the graphical model shown in Figure 2.4(b). In this figure rounded rectangle nodes are used to denote deterministic quantities and shaded circular nodes indicate what is observed in $\mathcal{D}$.

Given observed data, $\mathcal{D}$, one can then calculate the posterior on Θ. A second approach to deal with the unknown parameters is to find the maximum a posteriori (MAP) Θ using this posterior:

$$\Theta^* = \arg\max_{\Theta} p\left(\Theta|\mathcal{D}\right) \tag{2.19}$$

$$= \arg\max_{\Theta} p\left(\mathcal{D}|\Theta\right) p_0\left(\Theta|\Upsilon\right) \tag{2.20}$$

Calculating the posterior and/or calculating the MAP estimate can be difficult depending on the form of the chosen prior $p_0\left(\Theta|\Upsilon\right)$. However, the optimization and general calculation of the posterior is simplified greatly when a *conjugate* prior exists and is used. A family of priors specified by $p_0\left(\Theta|\Upsilon\right)$ is *conjugate* if the posterior $p\left(\Theta|\mathcal{D},\Upsilon\right)$ remains in the family. That is, it is conjugate if $p\left(\Theta|\mathcal{D},\Upsilon\right) = p_0\left(\Theta|\bar{\Upsilon}\right)$ and $\bar{\Upsilon}$ is a function of $\mathcal{D}$ and the original Υ.

A third approach is to marginalize over the parameters Θ and use the evidence when calculating probabilities:

$$p\left(\mathcal{D}\right) = \int_{\Theta} p\left(\mathcal{D}|\theta\right) p_0\left(\Theta|\Upsilon\right) d\Theta \tag{2.21}$$

Integrating over the space of all parameters is difficult in general. However, again, having a *conjugate* prior allows for tractable computation since the evidence can be written as:

$$p\left(\mathcal{D}\right) = \frac{p\left(\mathcal{D}|\Theta\right) p_0\left(\Theta|\Upsilon\right)}{p\left(\Theta|\mathcal{D},\Upsilon\right)} = \frac{p\left(\mathcal{D}|\Theta\right) p_0\left(\Theta|\Upsilon\right)}{p_0\left(\Theta|\bar{\Upsilon}\right)} \tag{2.22}$$

In the following sections we will overview specific parameterized families of distributions, ML estimates of their parameters, corresponding conjugate priors and detail how

to calculate evidence. Each of the distributions presented belongs to the *exponential family* of distributions [3].

■ 2.4.1 Discrete Distribution

Let $\mathbf{x}$ be a discrete random variable taking on one of K possible values in the set $\{1, \ldots, K\}$. A discrete distribution is a probability mass function with probability π_k that $\mathbf{x}$ takes on value k:

$$p(\mathbf{x}|\Theta) = \text{Discrete}(\mathbf{x}; \pi) \tag{2.23}$$

$$= \pi_{\mathbf{x}} \tag{2.24}$$

$$= \prod_{k=1}^{K} \pi_k^{\delta(\mathbf{x}-k)} \tag{2.25}$$

where $\Theta = \{\pi_1, \ldots, \pi_K\}$ and $\delta(u)$ is the discrete delta function taking a value of 1 if $u = 0$ and 0 otherwise. Given N samples of $\mathbf{x}$ as $\mathcal{D} = \{\mathcal{D}_1, \ldots, \mathcal{D}_N\}$, the ML estimate of Θ is simply:

$$\hat{\Theta} = \arg\max_{\Theta} p(\mathcal{D}|\Theta) = \{\hat{\pi}_1, \ldots, \hat{\pi}_K\} = \left\{\frac{n_1}{N}, \ldots, \frac{n_k}{N}\right\} \tag{2.26}$$

where n_k is a count of the number data points which had value k, $|\{\ i \mid \mathcal{D}_i = k\}|$.

Dirichlet Distribution

The conjugate prior for a discrete distribution is the *Dirichlet* distribution[1]. Given hyperparameters $\Upsilon = \alpha = \{\alpha_1, \ldots, \alpha_k\}$ the Dirchlet distribution has the form:

$$p_0(\Theta|\Upsilon) = p(\pi|\alpha) = \text{Dir}(\pi_1, \ldots, \pi_K; \alpha_1, \ldots, \alpha_k) \tag{2.27}$$

$$= \frac{\Gamma\left(\sum_k \alpha_k\right)}{\prod_k \Gamma(\alpha_k)} \prod_{k=1}^{K} \pi_k^{\alpha_k - 1} \tag{2.28}$$

[1]The Dirichlet distribution is normally said to be conjugate for the multinomial distribution. Given N discrete random variables each taking K values, the multinomial is a distribution on the counts n_k rather than the particular sequence. The Dirichlet is also conjugate for the simple discrete distribution, used here, which models the sequence of N independent observations rather than just the counts.

A uniform distribution can be obtained by setting all $\alpha_k = 1$. It is simple to see why this distribution is conjugate. The posterior on the parameters Θ, given data is:

$$p\left(\Theta|\mathcal{D},\Upsilon\right) \propto p\left(\mathcal{D}|\Theta\right) p_0\left(\Theta|\Upsilon\right) \tag{2.29}$$

$$\propto \prod_{k=1}^{K} \pi^{\alpha_k + n_k - 1} \tag{2.30}$$

$$\propto \mathrm{Dir}\left(\Theta; \alpha_1 + n_1, \ldots, \alpha_K + n_K\right) \tag{2.31}$$

or can equivalently be written as $p\left(\Theta|\mathcal{D},\Upsilon\right) = p_0\left(\Theta|\bar{\Upsilon}\right)$ where $\bar{\Upsilon} = \{\alpha_1 + n_1, \ldots, \alpha_K + n_K\}$. The evidence is simply:

$$p\left(\mathcal{D}|\Upsilon\right) = \frac{p\left(\mathcal{D}|\Theta\right) p_0\left(\Theta|\Upsilon\right)}{p\left(\Theta|\mathcal{D},\Upsilon\right)} \tag{2.32}$$

$$= \frac{\Gamma\left(\sum_k \alpha_k\right) \prod_k \Gamma\left(\alpha_k + n_k\right)}{\Gamma\left(N + \sum_k \alpha_k\right) \prod_k \Gamma\left(\alpha_k\right)} \tag{2.33}$$

The hyperparameters α_k can be thought of as a prior/pseudo count of how many times one saw the value K in $\sum_k \alpha_k$ trials. Figure 2.5 shows several Dirichlet distributions using different hyperparameters.

■ 2.4.2 Normal Distribution

Perhaps the most commonly used distrib ution for continuous valued random vectors $\mathbf{x}$ whose values are in R^d is the *Gaussian* or *normal* distribution. Given parameters $\Theta = \{\mu, \Lambda\}$ it takes the form:

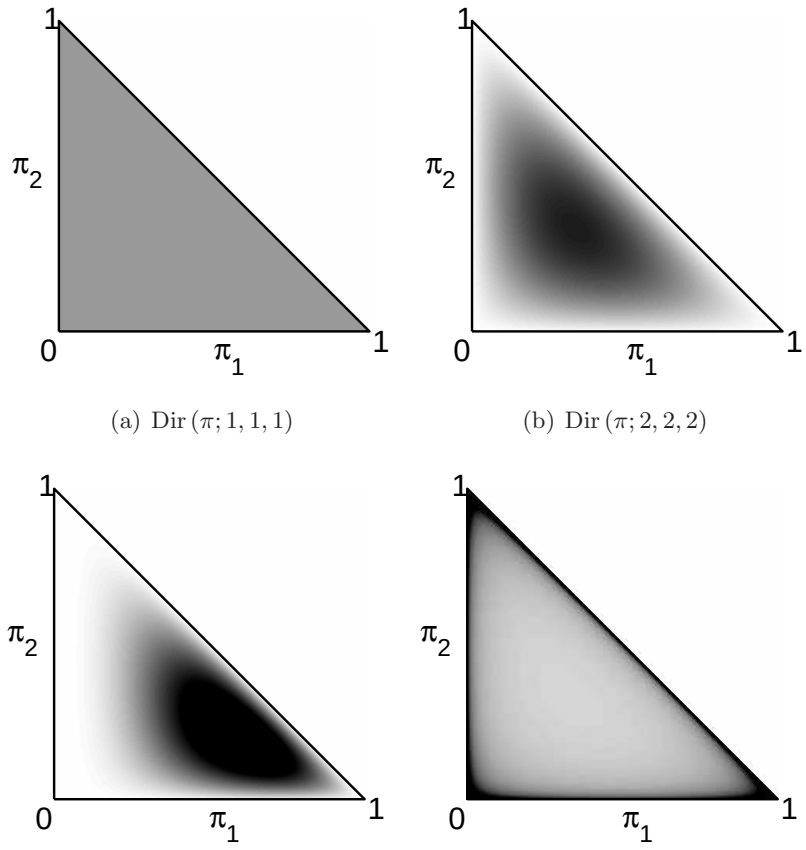
$$p\left(\mathbf{x}|\Theta\right) = \mathcal{N}\left(\mathbf{x}; \mu, \Lambda\right) \tag{2.34}$$

$$= \frac{1}{(2\pi)^{d/2}|\Lambda|^{1/2}} \exp\left\{\frac{1}{2}(\mathbf{x}-\mu)^{\mathsf{T}}\Lambda^{-1}(\mathbf{x}-\mu)\right\} \tag{2.35}$$

where $\mu \in R^d$ is the mean and $\Lambda \in R^{d\times d}$ is a $d \times d$ positive definite matrix representing the covariance among the elements in $\mathbf{x}$.

Given N independent samples of $\mathbf{x}$ in $\mathcal{D}$, the ML estimate of the Gaussian's parameters are the sample mean and covariance:

$$\hat{\mu} = \frac{1}{N}\sum_{n=1}^{N} \mathcal{D}_n \ , \qquad \hat{\Lambda} = \frac{1}{N}\sum_{n=1}^{N} (\mathcal{D}_n - \hat{\mu})(\mathcal{D}_n - \hat{\mu})^{\mathsf{T}} \tag{2.36}$$



(a) Dir $(\pi; 1, 1, 1)$ (b) Dir $(\pi; 2, 2, 2)$

(c) Dir $(\pi; 4, 2, 2)$ (d) Dir $(\pi; .5, .5, .5)$

Figure 2.5. *Example Dirichlet Distributions:* Distributions for $K = 3$ are displayed on the simplex $(\pi_1, \pi_2, 1 - \pi_1 - \pi_2)$. Dark represents high probability. (a) Uniform prior, (b) Prior biased toward equal π_ks (c) Prior biased toward $K = 1$, (c) By setting $\alpha_k < 0$ one obtains a prior that favors a sparse π.

Normal-Inverse-Wishart Distribution

The conjugate prior for the normal distribution is the *normal-inverse-Wishart* distribution. It factorizes as a normal prior on the mean given the covariance and an inverse-Wishart prior on the covariance given hyperparameters $\Upsilon = \{\omega, \kappa, \Xi, \nu\}$:

$$p_0(\Theta|\Upsilon) = p_0(\mu|\Lambda, \Upsilon)\, p_0(\Lambda|\Upsilon) \tag{2.37}$$

$$= \mathcal{N}(\mu; \omega, \Lambda/\kappa)\, \mathcal{IW}(\Lambda; \Xi, \nu) \tag{2.38}$$

Here the conditional prior on μ is a Gaussian with an expected value of $\omega \in R^d$ and covariance scaled by $\kappa \in R^1$. The hyperparameter κ can be thought of a count of prior "pseudo-observations." The higher κ is the tighter the prior on μ becomes around its mean ω. The d-dimensional inverse-Wishart distribution with positive definite parameter $\Xi \in R^{d \times d}$ and $\nu \in R^1$ degrees of freedom has the form:

$$\mathcal{IW}(\Lambda; \Xi, \nu) = \frac{|\Xi|^{\nu/2}|\Lambda|^{-\frac{(d+\nu+1)}{2}} \exp\left\{-\frac{1}{2}\text{tr}\left(\Xi\Lambda^{-1}\right)\right\}}{2^{\frac{\nu d}{2}}\pi^{d(d-1)/4}\prod_{i=1}^{d}\Gamma\left(\frac{\nu+1-i}{2}\right)} \tag{2.39}$$

The expected value of this distribution is simply $\Xi/(\nu - d - 1)$. It can be shown that, given N independent Gaussian observations the posterior distribution has the form:

$$p(\Theta|\mathcal{D}, \Upsilon) = \mathcal{N}(\mu; \bar{\omega}, \Lambda/\bar{\kappa})\,\mathcal{IW}\left(\Lambda; \bar{\Xi}, \bar{\nu}\right) \tag{2.40}$$

where the updated hyperparameters are

$$\bar{\kappa} = \kappa + N \tag{2.41}$$

$$\bar{\nu} = \nu + N \tag{2.42}$$

$$\bar{\omega} = \frac{\kappa}{\kappa + N}\omega + \frac{1}{\kappa + N}\sum_{n=1}^{N}\mathcal{D}_n \tag{2.43}$$

$$\bar{\Xi} = \Xi + \sum_{n=1}^{N}\mathcal{D}_n\mathcal{D}_n^{\mathsf{T}} + \kappa\omega\omega^{\mathsf{T}} - \bar{\omega}\bar{\omega}^{\mathsf{T}}. \tag{2.44}$$

Figure 2.6 shows an example normal-inverse-Wishart prior and an associated posterior given samples drawn from a normal distribution.

Integrating out parameters, the evidence takes the form:

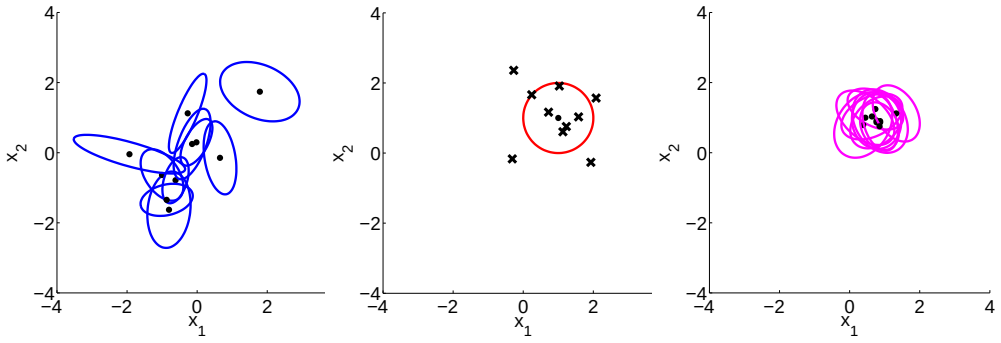
$$p(\mathcal{D}|\Upsilon) = \frac{\prod_{i=1}^{d}\Gamma\left((\nu + N + 1 - i)/2\right)}{\prod_{i=1}^{d}\Gamma\left((\nu + 1 - i)/2\right)}\left(\frac{\kappa}{\kappa + N}\right)^{d/2}\frac{|\Xi|^{\nu/2}}{\pi^{Nd/2}|\bar{\Xi}|^{(\nu+N)/2}} \tag{2.45}$$

This is an evaluation of a *multivariate-T* distribution. We will show this is a special case of the more general *matrix-T* distribution in the next section.

■ 2.4.3 Matrix-Normal Distribution

Next consider the case when one needs to model a conditional distribution $p(\mathbf{y}|\mathbf{x})$ where $\mathbf{y} \in R^d$ and $\mathbf{x} \in R^m$. The normal distribution can be generalized to model a linear Gaussian relationship such that

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{e}, \tag{2.46}$$



(a) Prior (b) Observed (c) Posterior

Figure 2.6. *Examples of the Normal-Inverse-Wishart distribution:* (a) Shows 10 2-d normal distributions with their parameters samples from a normal-inverse-wishart prior $\mathcal{N}(\mu; \mathbf{0}, 2\Lambda)\, \mathcal{IW}(\Lambda; \mathbf{I}_2, 5)$. Black dots represent the mean and the covariance is plotted as an ellipse. (b) Samples from a normal distribution with mean $[1\ 1]^\mathsf{T}$ and covariance $\mathbf{I}_2$. (c) Gaussians with their parameters sampled from the posterior on parameters given the samples in (b).

where $\mathbf{A} \in R^{d \times m}$ and the random variable $\mathbf{e} \in R^d$ is drawn from a zero mean normal distribution with covariance $\mathbf{\Lambda}$. The conditional distribution is thus

$$p(\mathbf{y}|\mathbf{x}, \Theta) = \mathcal{N}(\mathbf{y}; \mathbf{A}\mathbf{x}, \Lambda). \tag{2.47}$$

Under this model the ML parameter estimates given N observations of $\mathbf{y}$ and $\mathbf{x}$, $\mathcal{D}^y$ and $\mathcal{D}^x$ are

$$\hat{\mathbf{A}} = \left(\sum_{n=1}^{N} \mathcal{D}_n^y \mathcal{D}_n^{x\mathsf{T}}\right) \left(\sum_{n=1}^{N} \mathcal{D}_n^x D_n^{x\mathsf{T}}\right)^{-1} \tag{2.48}$$

$$\hat{\Lambda} = \frac{1}{N} \sum_{n=1}^{N} \left(\mathcal{D}_n^y - \hat{A}\mathcal{D}_n^x\right) \left(\mathcal{D}_n^y - \hat{A}\mathcal{D}_n^x\right)^\mathsf{T} \tag{2.49}$$

Matrix-Normal-Inverse-Wishart Distribution

The conjugate prior for this model is a generalization of the normal-inverse-Wishart distribution. Given $\Upsilon = \{\Omega, \mathbf{K}, \Xi, \mathbf{K}, \nu\}$ a *matrix-normal-inverse-Wishart* distribution has the form

$$p_0(\Theta|\Upsilon) = \mathcal{MN}(\mathbf{A}; \Omega, \Lambda, \mathbf{K})\, \mathcal{IW}(\Lambda; \Xi, \nu), \tag{2.50}$$

where the matrix-normal distribution is

$$\mathcal{MN}\left(\mathbf{A};\Omega,\Lambda,\mathbf{K}\right)=\frac{|\mathbf{K}|^{d/2}}{|2\pi\Lambda|^{m/2}}\exp\left\{-\frac{1}{2}\mathrm{tr}\left((\mathbf{A}-\boldsymbol{\Omega})^{\mathsf{T}}\Lambda^{-1}(\mathbf{A}-\Omega)\mathbf{K}\right)\right\} \tag{2.51}$$

and $\Omega\in R^{d\times m}$, $\mathbf{K}\in R^{m\times m}$. If $\mathrm{vec}\left(\mathbf{A}\right)$ represents the columns of $\mathbf{A}$ stacked into a vector, then it is normally distributed as

$$p\left(\mathrm{vec}\left(\mathbf{A}\right)|\Omega,\Lambda,\mathbf{K}\right)=\mathcal{N}\left(\mathrm{vec}\left(\mathbf{A}\right);\mathrm{vec}\left(\Omega\right),\Lambda\otimes\mathbf{K}^{-1}\right) \tag{2.52}$$

where $\otimes$ is the Kronecker product.

Given N independent observations, the posterior is

$$p\left(\Theta|\mathcal{D}^{y},\mathcal{D}^{x},\Upsilon\right)=\mathcal{MN}\left(\mathbf{A};\bar{\Omega},\Lambda,\bar{\mathbf{K}}\right)\mathcal{IW}\left(\Lambda;\bar{\Xi},\bar{\nu}\right) \tag{2.53}$$

where

$$\bar{\kappa}=\kappa+N \qquad\qquad \bar{\nu}=\nu+N \tag{2.54}$$

$$\bar{\Omega}=\Sigma_{y,x}\Sigma_{x,x}^{-1} \qquad\qquad \bar{\Xi}=\Xi+\Sigma_{y|x} \tag{2.55}$$

with

$$\Sigma_{x,x}=\sum_{n=1}^{N}\mathcal{D}_{n}^{x}\mathcal{D}_{n}^{x\mathsf{T}}+\mathbf{K} \qquad\qquad \Sigma_{y,x}=\sum_{n=1}^{N}\mathcal{D}_{n}^{y}\mathcal{D}_{n}^{x\mathsf{T}}+\Omega\mathbf{K} \tag{2.56}$$

$$\Sigma_{y,y}=\sum_{n=1}^{N}\mathcal{D}_{n}^{y}\mathcal{D}_{n}^{y\mathsf{T}}+\Omega\mathbf{K}\Omega^{\mathsf{T}} \qquad\qquad \Sigma_{y|x}=\Sigma_{y,y}-\Sigma_{y,x}\Sigma_{x,x}^{-1}\Sigma_{y,x}{}^{\mathsf{T}} \tag{2.57}$$

The evidence is calculated via

$$p\left(\mathcal{D}^{y}|\mathcal{D}^{x},\Upsilon\right)=\frac{\prod_{i=1}^{d}\Gamma\left((\nu+N+1-i)/2\right)}{\prod_{i=1}^{d}\Gamma\left((\nu+1-i)/2\right)}\frac{|\mathbf{K}|^{d/2}}{|\Sigma_{x,x}|^{d/2}}\frac{|\Xi|^{\nu/2}}{\pi^{Nd/2}|\bar{\Xi}|^{(\nu+N)/2}} \tag{2.58}$$

$$=\mathcal{MT}\left(\mathcal{D}^{y};\Omega\mathcal{D}^{x},\Xi,\mathbf{I}_{N}-\mathcal{D}^{x\mathsf{T}}\Sigma_{x,x}^{-1}\mathcal{D}^{x},\nu+N\right) \tag{2.59}$$

where $\mathcal{D}^{x}$ is being used as a $m\times n$ matrix, $\mathbf{I}_{k}$ represents a $k\times k$ identity matrix and $\mathcal{MT}\left(\mathbf{A};\mathbf{M},\mathbf{V},\mathbf{K},n\right)$ is a *matrix-T* distribution which has the form:

$$\frac{\prod_{i=1}^{d}\Gamma\left((n+1-i)/2\right)}{\prod_{i=1}^{d}\Gamma\left((n-m+1-i)/2\right)}\frac{|\mathbf{K}|^{d/2}}{|\pi\mathbf{V}|^{m/2}}\left|(\mathbf{A}-\mathbf{M})^{\mathsf{T}}\mathbf{V}^{-1}(\mathbf{A}-\mathbf{M})\mathbf{K}+\mathbf{I}_{m}\right|^{n/2} \tag{2.60}$$

Note that one obtains a normal and corresponding normal-inverse-wishart distribution for the case in which $m=1$,$\mathbf{x}=1$ and $\mathbf{A}=\mu$.

2.5 Select Elements of Information Theory

Information theory provides tools for quantifying uncertainty and statistical dependence. In addition, it is closely connected to statistical inference and learning [59]. Here, we provide a quick overview of the select elements of information theory we will use in this dissertation. We point the reader to Cover and Thomas [21] for more details.

Shannon's *entropy* provides a measure of information and inherent uncertainty in a random variable. For a discrete random variable with K possible values and distribution $p(\mathbf{x})$, entropy is defined as

$$H(\mathbf{x}) = -\sum_{k=1}^{K} p(k) \log p(k) \tag{2.61}$$

The entropy is maximized when $p(\mathbf{x}) = 1/K$ is a uniform distribution, corresponding to the most uncertainty. For continuous random variables, an extension of this definition is that of *differential entropy*:

$$H(\mathbf{x}) = -\int p(\mathbf{x}) \log p(\mathbf{x}) d\mathbf{x} \tag{2.62}$$

Joint and conditional entropy can be defined similarly:

$$H(\mathbf{x}, \mathbf{y}) = -\int\int p(\mathbf{x}, \mathbf{y}) \log p(\mathbf{x}, \mathbf{y}) \, d\mathbf{x} d\mathbf{y} \tag{2.63}$$

$$H(\mathbf{y}|\mathbf{x}) = -\int\int p(\mathbf{x}, \mathbf{y}) \log p(\mathbf{y}|\mathbf{x}) \, d\mathbf{x} d\mathbf{y} \tag{2.64}$$

It is important to note that conditional entropy is obtained via an expectation over the full joint. It can alternatively be defined in terms of an expectation of the entropy of one variable given a particular value of the other.

$$H(\mathbf{y}|\mathbf{x}) = \mathbb{E}_{p(\mathbf{x})} \left[H(\mathbf{y}|\mathbf{x} = x) \right] \tag{2.65}$$

$$= \int p(\mathbf{x} = x) \, H(\mathbf{y}|\mathbf{x} = x) \, dx \tag{2.66}$$

$$= -\int p(\mathbf{x} = x) \int p(\mathbf{y}|\mathbf{x} = x) \log p(\mathbf{y}|\mathbf{x} = x) \, d\mathbf{y} dx \tag{2.67}$$

$$= -\int\int p(\mathbf{x}, \mathbf{y}) \log p(\mathbf{y}|\mathbf{x}) \, d\mathbf{y} d\mathbf{x} \tag{2.68}$$

With these definitions it is straightforward to show that the joint uncertainty can be expressed as:

$$H(\mathbf{x}, \mathbf{y}) = H(\mathbf{x}) + H(\mathbf{y}|\mathbf{x}) \tag{2.69}$$

and that conditioning is guarantied not to increase uncertainty:

$$H(\mathbf{y}) \geq H(\mathbf{y}|\mathbf{x}) \tag{2.70}$$

Using this property one obtains a simple understanding of *mutual information* (MI):

$$I(\mathbf{x};\mathbf{y}) = H(\mathbf{y}) - H(\mathbf{y}|\mathbf{x}) \tag{2.71}$$

$$= H(\mathbf{x}) - H(\mathbf{x}|\mathbf{y}) \tag{2.72}$$

That is, MI measures the decrease in uncertainty in one random variable when conditioning on the other. For independent random variables $p(\mathbf{y}|\mathbf{x}) = p(\mathbf{y})$, thus $H(\mathbf{y}|\mathbf{x}) = H(\mathbf{y})$, making the MI go to 0.

MI can be expressed as a *Kullback-Leibler (KL) divergence*. KL divergence measure the *relative entropy* between two probability distributions $p(\mathbf{x})$ and $q(\mathbf{x})$:

$$D\left(p(\mathbf{x}) \,\middle\|\, q(\mathbf{x}) \right) = \int p(\mathbf{x}) \log \frac{p(\mathbf{x})}{q(\mathbf{x})} d\mathbf{x} \tag{2.73}$$

$$= \mathbb{E}_{p(\mathbf{x})}\left[\log \frac{p(\mathbf{x})}{q(\mathbf{x})} \right] \tag{2.74}$$

Similar to conditional entropy, the KL divergence between two conditional distributions is:

$$D\left(p(\mathbf{y}|\mathbf{x}) \,\middle\|\, q(\mathbf{y}|\mathbf{x}) \right) = \int p(\mathbf{x},\mathbf{y}) \log \frac{p(\mathbf{y}|\mathbf{x})}{q(\mathbf{y}|\mathbf{x})} d\mathbf{y} d\mathbf{x} \tag{2.75}$$

Consider a binary hypothesis test for choosing a distribution on $\mathbf{x}$ in which hypothesis H_1 and H_2 are defined as:

$$H_1 : \mathbf{x} \sim p(\mathbf{x}) \tag{2.76}$$

$$H_2 : \mathbf{x} \sim q(\mathbf{x}) \tag{2.77}$$

Assuming equation priors, $p(H_1) = p(H_2)$, a log likelihood ratio test takes the form:

$$l_{1,2} \triangleq \log \frac{p(\mathbf{x})}{q(\mathbf{x})} \begin{array}{c} H_1 \\ > \\ \leq \\ H_2 \end{array} 0 \tag{2.78}$$

It is easy to see the intimate connection between this hypothesis test and KL divergence. In expectation under H_1 the log likelihood ratio is simply a measure of relative entropy.

Two alternative views of MI emerge. It can be thought of as the KL divergence between the joint distribution of two random variables and the product of their marginals,

$$I(\mathbf{x};\mathbf{y}) = D\left(\; p(\mathbf{x},\mathbf{y}) \; \middle\| \; p(\mathbf{x})\, p(\mathbf{y}) \; \right) \tag{2.79}$$

or simply in terms of a hypothesis test on the presence of statistical dependence.

■ 2.6 Summary

We defined statistical dependence and discussed how it can be encoded in a graphical model. Markov models for time-series were presented as stochastic processes with a fixed temporal dependence structure. Additionally, we presented a select set of parametric families of marginal and conditional distributions, along with conjugate priors which allow for efficient learning, inference and evidence calculations. Lastly, we briefly overviewed information theory and showed how entropy and KL divergence can be used to provide a measure of uncertainty and statistical dependence.

This dissertation will build upon the basic foundation this chapter has provided. In the next chapter we present graphical models for describing the dependence among multiple time-series. We focus on understanding this dependence given a set of observations and use connections to information theory to characterize the difficulty of a structural inference task. In addition, we will show how conjugate priors allow for tractable inference and characterization of uncertainty in structure.

Chapter 3

Static Dependence Models for Time-Series

In this chapter we present the underlying probabilistic models we use to describe dependent time-series. While our ultimate aim is to perform dependence analysis in a dynamic setting, it is important to first understand the more restricted static case. Given observations of multiple time-series over a set of times, the goal of a *static dependence analysis* task is to describe the relationships among these observed time-series in terms of a fixed dependence structure. Here, we examine static dependence analysis in detail and cast the problem as inference over structure in a *static dependence model*. A *static dependence model* is a probabilistic model which describes the evolution of multiple time-series in addition to the relationships among them in terms of a common dependence structure over all time. It is important to note that, while the framework developed here can be used to learn predictive models for classification and examination of time-series observed in the future, our primary focus is on structure discovery as a tool for data analysis.

We begin by establishing some standard notation which will be used for the remainder of the dissertation. A general static dependence model for time-series is presented along with a discussion of the key challenges for inference using these models. Next, in Section 3.3 a specific static dependence model is introduced which describes the dependence among multiple time-series in terms of sets of independent groups. Using this model, we focus on cases in which one is interested in identifying one active structure from a relatively small (and therefore tractable) enumerated set of possible dependence relationships. In the absence of any prior knowledge, inference over structure is discussed in a maximum likelihood framework and shown to be a point estimate approximation to Bayesian inference.

Moving beyond a tractable enumerated set of structures, in Section 3.4, we present a more detailed directed static dependence model which explicitly encodes the causal relationships among time-series in terms of directed structures. The number of possible directed structures grows super-exponentially with the number of time-series being modeled. A conjugate prior on the directed structure and parameters of this model is presented which allows one to reason over the set of all directed structures in exponential-time complexity. Furthermore, by imposing simple local or global structural constraints we show that one can reduce the exponential-time complexity to polynomial-time complexity for reasoning over a still super-exponential number of candidate structures. Specifically we focus on bounded in-degree structures with directed trees and forests being special cases with global constraints. These constraints yield tractable Bayesian inference over directed structures, allowing exact calculation of the partition function as well as additional marginal event probabilities. The method we present for Bayesian reasoning over structure is closely related to that of [33, 54], but extended to the analysis of time-series for which the strict temporal ordering provides a computational advantage.

■ 3.1 Notation

We begin by introducing some notation for the purpose of explicitly denoting individual time-series, past values of individual time-series as well as sets thereof. This notation is summarized in Table 3.1 for future reference.

Consider N time-series and let $\mathbf{x}_t^v$ be a random variable representing the vector value of the v-th time-series at time t. The random variable $\mathbf{X}_t$ is the set of the random variables for all N time-series at time t, *i.e.* $\mathbf{X}_t = \left\{\mathbf{x}_t^1, \ldots, \mathbf{x}_t^N\right\}$. Subsets $\mathbf{S} \subset \{1, \ldots, N\}$ of time-series can be indexed using the notation $\mathbf{x}_t^{\mathbf{S}}$. For example, the random variable representing values of times-series 1,2 and 4 at time t is $\mathbf{x}_t^{1,2,4}$. For a model of order r, the r past values of the v-th time-series at time t is represented by the random variable $\widetilde{\mathbf{x}}_t^v$. That is, $\widetilde{\mathbf{x}}_t^v$ is the set of $\left\{\mathbf{x}_{t-1}^v, \ldots, \mathbf{x}_{t-r}^v\right\}$. As we will be explicit on the temporal model order, r is suppressed in the notation, $\widetilde{\mathbf{x}}_t^v$, for brevity. The random variable $\widetilde{\mathbf{X}}_t = \widetilde{\mathbf{x}}_t^{1,\ldots,N} = \left\{\widetilde{\mathbf{x}}_t^1, \ldots, \widetilde{\mathbf{x}}_t^N\right\}$ is the r past values for all N time-series.

Multiple time points can be indexed by a vector $\mathbf{t} = [t_1, t_2, ..., t_T]$ such that $\mathbf{X_t} = \{\mathbf{X}_{t_1}, ..., \mathbf{X}_{t_T}\}$. We will sometimes reference a continuous set of time using the notation $1 : T$ rather than $\mathbf{t} = \{1, 2, \ldots, T\}$. Note that the full past $\widetilde{\mathbf{X}}_\mathbf{t}$ can be formed from

N	Number of time-series
r	Order of Markov temporal dependence. The current time is dependent on the r past values.
$\mathbf{x}_t^v$	Random variable representing the vector value of the v-th time-series at time t
$\mathbf{x}_t^{\mathbf{S}} = \mathbf{x}_t^{S_1,\ldots,S_m}$	Random variable representing a set of time-series at time t where $\mathbf{S} = \{S_1, \ldots, S_m\}$. For example, $\mathbf{x}_t^{1,2}$ represents $\mathbf{x}_t^1$ and $\mathbf{x}_t^2$. We will sometimes treat $\mathbf{x}_t^{\mathbf{S}}$ as a single random vector.
$\widetilde{\mathbf{x}}_t^v$	Random variable representing the r past of time-series v at time t. Specifically, $\widetilde{\mathbf{x}}_t^v$ represents the set $\widetilde{\mathbf{x}}_{t-1}^v$ through $\widetilde{\mathbf{x}}_{t-r}^v$. Three simple cases: If $r = 0$, $\widetilde{\mathbf{x}}_t^v = \emptyset$. If $r = 1$, $\widetilde{\mathbf{x}}_t^v = \mathbf{x}_{t-1}^v$. If $r = 2$, $\widetilde{\mathbf{x}}_t^v = \left\{\mathbf{x}_{t-1}^v, \mathbf{x}_{t-2}^v\right\}$
$\mathbf{X}_t$	Random variable representing all N time-series at time t. $\mathbf{X}_t = \mathbf{x}_t^{1,\ldots,N}$.
$\widetilde{\mathbf{X}}_t$	Random variable represent the r past of all N time-series at time t. $\widetilde{\mathbf{X}}_t = \widetilde{\mathbf{x}}_t^{1,\ldots,N}$.
$\mathbf{t} = \{t_1, t_2, ..., t_T\}$	A set of time points.
$1:T$	The set of numbers from 1 to T. In general, $a:b$ represents the set of numbers from a to b. Often we use this notation in place of $\mathbf{t} = \{1, \ldots, T\}$.
$\mathbf{X_t}$	Random variable representing all time-series at the time points specified by $\mathbf{t}$, $\{\mathbf{X}_{t_1}, \ldots, \mathbf{X}_{t_T}\}$. Similarly $\mathbf{x}_{\mathbf{t}}^{\mathbf{S}}$ is all time-series specified in set $\mathbf{S}$ at the points specified in $\mathbf{t}$.
$\mathcal{D}$	Observations of all time-series over all time.
$\mathcal{D}_t^{\mathbf{S}}$	Observation of $\mathbf{x}_t^{\mathbf{S}}$
$\widetilde{\mathcal{D}}$	Observation of all past values $\widetilde{\mathbf{X}}_{\mathbf{t}}$. Note that $\widetilde{\mathcal{D}}$ can be formed from $\mathcal{D}$ and additional information $\mathcal{C}$.
$\mathcal{C}$	Set of initial conditions or extra observations needed to form $\widetilde{\mathbf{X}}_{\mathbf{t}}$ and/or $\widetilde{\mathcal{D}}$. For example, if $\mathbf{t} = \{1, \ldots, N\}$ and $r = 0$, one needs $\mathcal{C} = \mathbf{X}_0$ to form $\widetilde{\mathbf{X}}_{\mathbf{t}}$.

Table 3.1. *Notation Summary:* Notation for time-series, past values of time-series and sets thereof.

$\mathbf{X_t}$ and a set $\mathcal{C}$ containing values not available in $\mathbf{X_t}$. For example, if $r = 1$, $\widetilde{\mathbf{X}}_{1:T}$ can be formed from $\mathbf{X}_{1:T}$ and initial conditions $\mathcal{C} = \{\mathbf{X}_0\}$. We will often treat these sets of random variables as random vectors or matrices when convenient. For example, one can treat $\mathbf{x}^{1,2}_{1:T}$ as a matrix:

$$\mathbf{x}^{1,2}_{1:T} = \begin{bmatrix} \mathbf{x}^{1}_{1:T} \\ \mathbf{x}^{2}_{1:T} \end{bmatrix} = \begin{bmatrix} \mathbf{x}^{1,2}_{1} & \dots & \mathbf{x}^{1,2}_{T} \end{bmatrix} = \begin{bmatrix} \mathbf{x}^{1}_{1} & \dots & \mathbf{x}^{1}_{T} \\ \mathbf{x}^{2}_{1} & \dots & \mathbf{x}^{2}_{T} \end{bmatrix} \tag{3.1}$$

We extend this notation to describe observed data. Let $\mathcal{D}$ be a set of T complete observations, $\{\mathbf{X}_1, \dots, \mathbf{X}_T\}$. We use the notation $\mathcal{D}^{\mathbf{S}}$ to denote observations of specified set of time-series, $\{\mathbf{x}^{\mathbf{S}}_1, \dots, \mathbf{x}^{\mathbf{S}}_T\}$. An observation at time t of all time-series is denoted as $\mathcal{D}_t$ and $\mathcal{D}^v_t$ specifies the observation for v-th time-series at time t. We use the notation $\widetilde{\mathcal{D}}$ for observations of the past $\{\widetilde{\mathbf{X}}_1, ..., \widetilde{\mathbf{X}}_T\}$. It can be formed using $\mathcal{D}$ and past information or initial conditions $\mathcal{C}$.

■ 3.2 Static Dependence Models

In this chapter, we focus on the design of a static dependence model for N time-series. Given a specified structure E and set of parameters Θ, this model is assumed to be r-th order Markov:

$$p\left(\mathbf{X_t}|E, \Theta, \mathcal{C}\right) = \prod_{i=1}^{T} p\left(\mathbf{X}_{t_i}|\widetilde{\mathbf{X}}_{t_i}, E, \Theta\right). \tag{3.2}$$

That is, we assume a fixed temporal structure in which $\mathbf{X}_t$ is dependent on its r past values $\widetilde{\mathbf{X}}_t$. Here, the structure E represents the dependence **among** the N time-series assuming this fixed temporal Markov relationship. In order to simplify notation we will drop the $\mathcal{C}$ when it is clear from the context and use $p\left(\mathbf{X_t}|E, \Theta\right)$.

Note that if $r = 0$ then $\widetilde{\mathbf{X}}_t = \emptyset$ and one obtains a distributions that is independent and identically distributed over time:

$$p\left(\mathbf{X_t}|E, \Theta\right) = \prod_{i=1}^{T} p\left(\mathbf{X}_{t_i}|E, \Theta\right). \tag{3.3}$$

Figure 3.1 summarizes the notation and general form of a static dependence model for $N = 2$ time-series with $r = 1$. It shows three views. The upper left depicts the two time-series in an abstract graphical model. The dependence structure E specifies relationships among the time-series. In the figure E is left abstract as shaded blue regions. Collapsing the time-series into single $\mathbf{X}_t$ at each point in time reveals the Markov

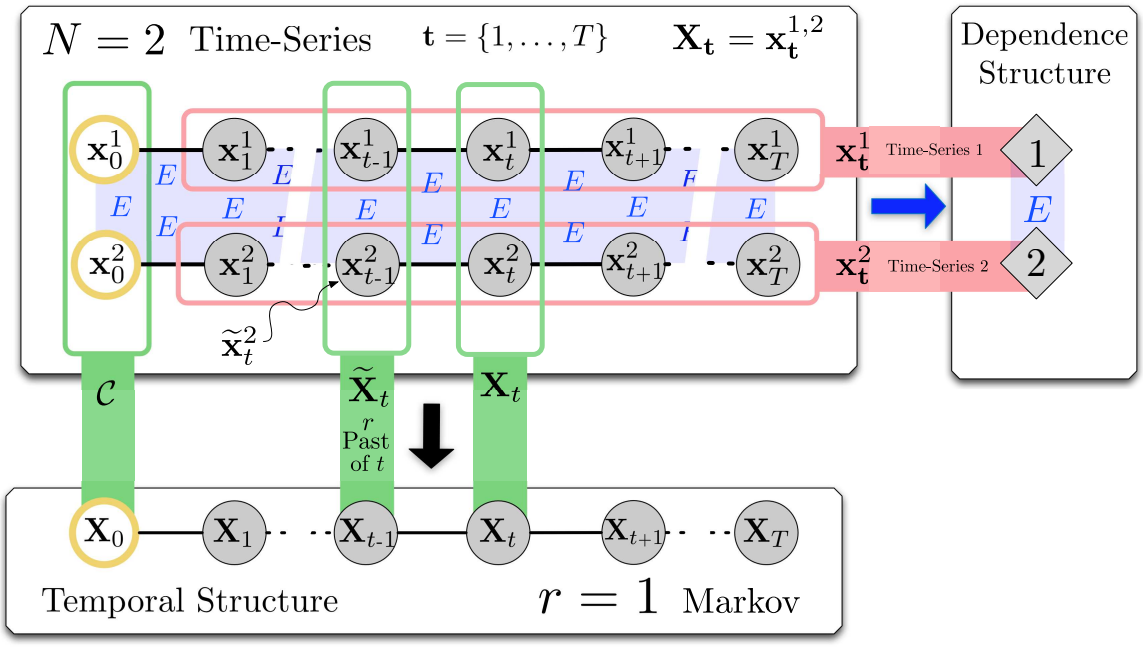


Figure 3.1. *Three Abstract Views of a Static Dependence Model:* Three views of a static dependence model for $N = 2$ time-series with accompanying notation are shown. The top left view shows an abstract undirected graphical model for the static dependence model in which the structure E would describe the relationships between the time-series in the shaded blue region. The bottom left view exposes the the r-th order temporal structure by representing both time-series as a single random variable at each t. The top right view collapses the time-series over all time and can be used to help understand the dependence structure between the time-series. Again, here E is left abstract.

structure shown in the temporal view on the bottom. The upper-right graph depicts each time-series collapsed over all time as a diamond shaped vertex. The dependence structure E specifies the relationship among these vertices in this *dependence graph*. We use diamond shaped vertices in the dependence graph so that it is not directly interpreted as a graphical model, but instead provides a summary of the dependence among time-series.

Concrete examples will be given in Sections 3.3 and 3.4 where we present two different static dependence models. The primary difference between the two models is the way in which the structure is specified and how we will use each model to reason about dependence among time-series. We will use the notation E when discussing a general static dependence model. This notation will change for each specific models in order

to directly reflect how their structure is specified. Specifically, in Section 3.3 we will introduce a model which specifies structure in terms of hyperedges F, while in Section 3.4 we will discuss a model which uses directed edges $\bar{E}$.

■ 3.2.1 Structural Inference

Given a static dependence model our ultimate goal is that of structural inference. That is, we are interested in inferring the structure E given a set of observed data $\mathcal{D}$ while treating Θ as nuisance parameters . In structure inference problems, as in any Bayesian inference problem, one would ideally like to calculate the posterior on structure given the observed data $\mathcal{D}$. Using Bayes rule and basic properties of probability we see that the posterior has the form:

$$\begin{aligned} p\left(E|\mathcal{D}\right) &= \frac{1}{p\left(\mathcal{D}\right)} p_0\left(E\right) p\left(\mathcal{D}|E\right) \\ &= \frac{1}{p\left(\mathcal{D}\right)} p_0\left(E\right) \int_{\Theta|E} p\left(\mathcal{D}|E,\Theta\right) p_0\left(\Theta|E\right) d\Theta. \end{aligned} \tag{3.4}$$

where $p_0\left(\right)$ is used to indicate a prior distribution. There are three main challenges one needs to address before being able to calculate the posterior:

1. One must define priors over the space of parameters and structures that can be reasoned over efficiently.

2. One must be able to tractably compute the integral over the unknown parameters.

3. In order to obtain an exact posterior probability, the data evidence, $p\left(\mathcal{D}\right)$, must be calculated. This involves integrating out **both** parameters Θ and structure E.

The first challenge is that of specifying prior knowledge about parameters and structure. That is, Equation 3.4 uses a prior on structure, $p_0\left(E\right)$, in addition to a prior on parameters given a specific structure $p_0\left(\Theta|E\right)$. What makes defining these terms difficult in general is threefold. First, depending on how the class of structures is specified, the number of possible structures can be very large. In general the number of allowable structures is super-exponential in the number of time-series. Second, one must provide a prior on parameters for each possible structure. Each structure will generally have a different number of parameters, each with a different role. For example, a structure which describes independent time-series needs fewer parameters than those which encode more dependence. Third, there is the question of what should be done when no

prior knowledge is available. Specifically, while one may be able to define uniform priors on the discrete set of structures, placing an uninformative prior over the continuous space of parameters can be difficult. One may have to turn to placing very broad priors as an approximation.

We will address the first challenge in two different ways using the two specific dependence models described in the following sections. In Section 3.3, we focus on cases in which we consider a small set of allowable structures and, in the absence of any prior information, adopt a frequentist view by concentrating on the likelihood term $p(\mathcal{D}|E,\Theta)$ rather than the posterior. In Section 3.4, we introduce a model along with a tractable conjugate prior on both the parameters and directed structure describing causal relationships among time-series.

The second challenge in calculating Equation 3.4 is related to the fact that we are primarily interested in making probabilistic statements regarding the structure and are treating the parameters Θ as a nuisance. The integration over all parameters weighted by their prior avoids having to estimate or choose one particular parameter, however, this may be difficult to compute. We address this challenge in two different ways using the two models presented in Sections 3.3 and 3.4. In Section 3.3.2 we discuss maximum likelihood inference and show the sense in which it approximates this integration with a point estimate. In Section 3.4.3, using a different model, we show how to calculate this evidence term using a conjugate prior.

The third challenge in Equation 3.4 is that of calculating $p(\mathcal{D})$. This is of concern if one desires an exact calculation of the posterior. Maximum a posteriori and maximum likelihood estimates of E do not require this normalization term. However, if an exact posterior can be obtained, a wide variety of useful statistics and exact marginal posterior probabilities can be calculated. These quantities allow a full characterization of uncertainty in the structure. In Section 3.4.3 we present priors which are conjugate and allow for tractable exact posterior calculations.

■ 3.3 Factorization Model

In the previous section we discussed the general form of static dependence models for time-series. Here, we specialize the model such that the dependence among time-series described in terms of independent groups. We denote these models as static *factorization models*, FactMs, and will sometimes use the notation FactM(r) to explicitly specify

Table 3.2. *Example FactMs:* Simple examples showing a specific factorization F and the form of the resulting distribution with $N = 2$ and $r = 1$ or $r = 0$.

r	F	$p\left(\mathbf{X}_t|\widetilde{\mathbf{X}}_t, F, \Theta\right)$
1	$\{\{1\},\{2\}\}$	$p\left(\mathbf{x}_t^1|\mathbf{x}_{t-1}^1, \Theta_1\right) p\left(\mathbf{x}_t^2|\mathbf{x}_{t-1}^2, \Theta_2\right)$
1	$\{\{1,2\}\}$	$p\left(\mathbf{x}_t^2, \mathbf{x}_t^2|\mathbf{x}_{t-1}^1, \mathbf{x}_{t-1}^2, \Theta_{1,2}\right)$
0	$\{\{1\},\{2\}\}$	$p\left(\mathbf{x}_t^2|\Theta_1\right) p\left(\mathbf{x}_t^2|\Theta_2\right)$
0	$\{\{1,2\}\}$	$p\left(\mathbf{x}_t^2, \mathbf{x}_t^2|\Theta_{1,2}\right)$

the temporal order r. The independent groups are specified by a set of hyperedges F which form a partitioning of the set $\{1, \ldots, N\}$. Specifically, F is a set of $|F|$ hyperedges where each $F_i \in F$ is restricted such that:

$$\bigcup_{i=1}^{|F|} F_i = \{1, 2, \ldots, N\} \tag{3.5}$$

$$F_i \bigcap F_j = \emptyset \;\; \forall \;\; i \neq j \;\; \in \{1, 2, \ldots, N\}. \tag{3.6}$$

That is, the union of all hyperedges is the full set $\{1, \ldots, N\}$ and no hyperedges share elements. For $N = 2$, only $F = \{\{1\},\{2\}\}$ and $F = \{\{1,2\}\}$ are consistent with this definition.

Given F and parameters Θ, a FactM conditional distribution takes the form:

$$p\left(\mathbf{X}_t|\widetilde{\mathbf{X}}_t, F, \Theta\right) = \prod_{f=1}^{|F|} p\left(\mathbf{x}_t^{F_f}|\widetilde{\mathbf{x}}_t^{F_f}, \Theta_{F_f}\right). \tag{3.7}$$

That is, the conditional distribution factorizes according to F with each factor having its own set of parameters. Note the change in notation from the more general $p\left(\mathbf{X}_t|\widetilde{\mathbf{X}}_t, E, \Theta\right)$ in Equation 3.2 to this specific form $p\left(\mathbf{X}_t|\widetilde{\mathbf{X}}_t, F, \Theta\right)$. That is, in order to be explicit about how structure is specified and to be consistent with the notation in Chapter 2 we replace E with hyperedges F when describing a FactM. In this context, will also often refer to the set of hyperedges F as a *factorization* and $F_i \in F$ as the i-th *factor*.

Table 3.2 presents some simple examples for $N = 2$. Perhaps the simplest model to understand is the case in which $r = 0$. If one assumes a Gaussian form for each factor model, each parameter Θ_{F_f} contains the mean and covariance of the elements in $\mathbf{x}_t^{F_f}$. Thus, the difference between the last two rows in Table 3.2 is that of a Gaussian with

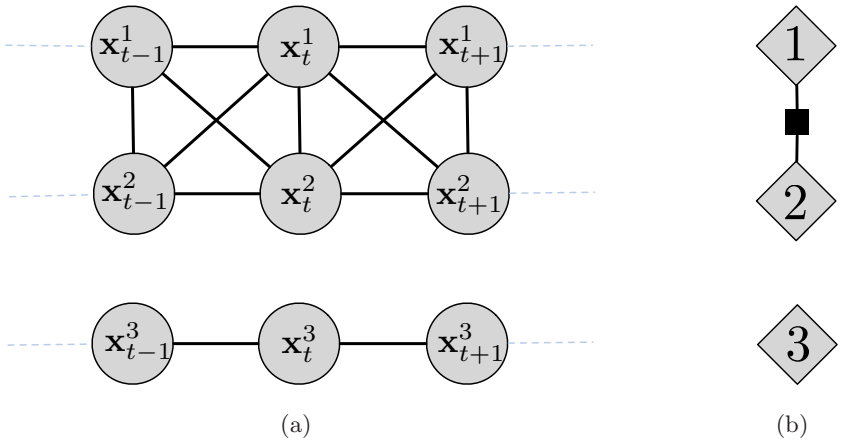


Figure 3.2. *Example FactM(1) Structure and Association Graph:* The Markov structure of an example FactM(1) model for $N = 3$ time-series is shown in (a) and corresponding association graph in the form of a factor graph is shown in (b).

a block diagonal covariance formed by the two block elements specified by Θ_1 and Θ_2 (3rd row), versus one with a full covariance specified by $\Theta_{1,2}$ (4th row).

Figure 3.2 shows two views of a FactM(1) for $N = 3$ time-series with factorization $F = \{\{1, 2\}, \{3\}\}$. This factorization is just one of the five possible for three time-series. Figure 3.2(a) shows the structure of this FactM as an undirected graphical model. Figure 3.2(b) provides an alternative view as a dependence factor graph. We will often refer to this factor graph as the *association graph* to emphasize that one can think of all time-series that belong to a common factor as being "associated". Additionally, we will use such graphs in Chapter 5 when describing an audio-visual association task. The association graph hides the temporal structure and uses diamond shaped vertices with a number to represent entire time-series.

■ 3.3.1 The Set of Factorizations

Definition 3.3.1 ($\mathcal{B}_N$): *$\mathcal{B}_N$ is the set of all partitions of the set $\{1, \ldots, N\}$*

As just discussed, the dependence structure of a FactM is specified by an $F \in \mathcal{B}_N$. An important question one may ask is: how many distinct dependence relationships can one represent? That is, what is the size of the set $\mathcal{B}_N$. This is a well understood

quantity called the Bell number[1] $B_N = |\mathcal{B}_N|$, [78]. For $N = 1$ there is only a single partition, $\{\{1\}\}$, and thus $B_1 = 1$. Higher Bell numbers can be obtained using the recursion

$$B_N = 1 + \sum_{k=1}^{N-1} \binom{N-1}{k} B_k \tag{3.8}$$

In words, the recursion says that to find the number of partitions of N numbers you sum up a series of terms from $k = 1$ to $k = N - 1$ each of which is the number of ways you can partition a set of k numbers, B_k, multiplied by the number of ways you could have chosen those k numbers from the a set of size $N-1$, $\binom{N-1}{k}$. The addition of 1 is due to the fact that there is always the partition which is the full set $\{\{1, \ldots, N\}\}$.

Equation 3.8 provides an algorithm for calculating the number of factorizations for a given N. Another important question is: how does this number grow with N? The asymptotic form of the Bell number was derived by De Bruijin [22] and simplifies to

$$\begin{aligned} B_N &= e^{N(\log N - \log\log N + O(1))} \\ &= N^N e^{N(O(1) - \log\log N)}. \end{aligned} \tag{3.9}$$

That is, the number of possible factorizations grows super-exponentially with N.

Reasoning over the full set, $\mathcal{B}_N$, is generally intractable for large N. However, in this dissertation, we use FactMs for tasks in which there is a small tractable subset of M factorizations one is interested in. Specifically, in Chapter 5 the factorization structures considered are linked to a tractable number of possible sources of speech in an audio-visual association task.

■ 3.3.2 Maximum-Likelihood Inference of FactM Structure

Given observed data $\mathcal{D}$, we wish to infer structure F. Here, using a FactM, we address the challenges discussed in Section 3.2.1 by adopting a frequentist view and focus on obtaining point estimates of structure rather than the full posterior. That is, in the absence of any prior information, we choose a maximum likelihood approach. Maximum likelihood (ML) seeks to find the structure which best explains the observed data. As discussed in Section 3.2.1 inference over structure requires some way of dealing with

[1] In honor of Eric Temple Bell

nuisance parameters Θ. A common approach in the ML framework is to use the max over all parameters as well. That is, for a general static dependence model with structure E we wish to find:

$$\hat{E} = \arg\max_{E} \max_{\Theta} p\left(\mathcal{D}|E,\Theta\right). \tag{3.10}$$

One can view this ML optimization as an approximation to finding the maximum a posterior (MAP) estimate of structure E. It is equivalent to setting a uniform prior on structure, $p_0\left(E\right) = \beta$, approximating the evidence $p\left(\mathcal{D}|E\right)$ with the a point estimate $p\left(\mathcal{D}|E,\hat{\Theta}\right)$ and maximizing Equation 3.4 over E:

$$\begin{aligned}
\arg\max_{E} p\left(E|\mathcal{D}\right) &= \arg\max_{E} p_0\left(E\right) \int_{\Theta|E} p\left(\mathcal{D}|E,\Theta\right) p_0\left(\Theta|E\right) d\Theta && (3.11)\\
&= \arg\max_{E} \beta \int_{\Theta|E} p\left(\mathcal{D}|E,\Theta\right) p_0\left(\Theta|E\right) d\Theta && (3.12)\\
&\approx \arg\max_{E} p\left(\mathcal{D}|E,\hat{\Theta}\right) && (3.13)\\
&\approx \arg\max_{E} \max_{\Theta} p\left(\mathcal{D}|E,\Theta\right) && (3.14)\\
&= \hat{E} && (3.15)
\end{aligned}$$

where the approximation made in Equation 3.13 is that the integral above can be well approximated by only considering a single point $\hat{\Theta}$. Equation 3.14 further assumes that the best single point is the one that maximizes $p\left(\mathcal{D}|E,\Theta\right)$.

We will use this ML approach with a FactM in an audio-visual association task in Chapter 5. Here, we examine this optimization in more detail specifically for the FactM. Substituting a FactM into Equation 3.10 and using the monotonicity of the log function one obtains

$$\begin{aligned}
\hat{F} &= \arg\max_{F} \max_{\Theta} p\left(\mathcal{D}|F,\Theta\right) && (3.16)\\
&= \arg\max_{F} \max_{\Theta} \frac{1}{T} \log p\left(\mathcal{D}|F,\Theta\right) && (3.17)\\
&= \arg\max_{F} \max_{\Theta} \frac{1}{T} \sum_{t=1}^{T} \sum_{f=1}^{|F|} \log p\left(\mathcal{D}_t^{F_f}|\widetilde{\mathcal{D}}_t^{F_f},\Theta_{F_f}\right) && (3.18)\\
&= \arg\max_{F} \sum_{f=1}^{|F|} \max_{\Theta_{F_f}} \frac{1}{T} \sum_{t=1}^{T} \log p\left(\mathcal{D}_t^{F_f}|\widetilde{\mathcal{D}}_t^{F_f},\Theta_{F_f}\right) && (3.19)
\end{aligned}$$

By simplifying notation, we see that ML inference of structure in a FactM involves finding the factorization F which maximizes the product over a set of weights associated with each factor:

$$\hat{F} = \arg\max_{F} \sum_{f=1}^{|F|} \hat{W}_{F_f} \tag{3.20}$$

where

$$\hat{W}_{F_f} \triangleq \max_{\Theta_{F_f}} \frac{1}{T} \sum_{t=1}^{T} \log p\left(\mathcal{D}_t^{F_f} | \widetilde{\mathcal{D}}_t^{F_f}, \Theta_{F_f}\right) \tag{3.21}$$

$$= \max_{\Theta_{F_f}} \hat{\mathbb{E}}\left[\log p\left(\mathcal{D}_t^{F_f} | \widetilde{\mathcal{D}}_t^{F_f}, \Theta_{F_f}\right)\right] \tag{3.22}$$

$$= -\min_{\Theta_{F_f}} \hat{H}\left(\mathcal{D}_t^{F_f} | \widetilde{\mathcal{D}}_t^{F_f}; \Theta_{F_f}\right) \tag{3.23}$$

Here $\hat{\mathbb{E}}\,[]$ is the sample average and $\hat{H}\,(;\theta)$ is the empirical estimate of entropy using the parameters θ. The weight, $\hat{W}_{F_f}$, is simply the log likelihood of the time-series indexed by factor F_f maximized over all parameters for that factor. This is related to finding the parameters which minimize the uncertainty, conditional entropy. For the simple case of $r = 0$ and Gaussian factors, the weight for factor F_f is the likelihood of the data within that factor, given the maximum likelihood estimate of the mean and covariance of that factor.

When considering all factorizations, there are $2^N - 1$ unique possible factor sets to calculate weights for. However, again, for our applications of interest we will restrict F to be one of M possible factorizations which potentially have many factors in common. This greatly reduces our search and the number of weights needed to be calculated.

A Closer Look at ML Inference of Structure

An alternative view of this particular ML inference task is as an M-ary hypothesis test over M allowable FactMs, each with a unique structure but unknown parameters. Here, we analyze how two hypothesized FactMs distinguish themselves from each other by examining the form of the likelihood ratio used when performing such a test. We will show that one can separate out the role of structure from that of parametric differences when deciding between two FactMs. When using an ML approach for estimating parameters from data, we show that separability due to parametric differences is lost,

making it more difficult to distinguish between hypothesized models. This form of analysis was originally presented by Ihler et al. [47]. Here, we specialize the analysis to our particular problem formulation. We contrast this analysis with inference using dynamic dependence models in Chapter 4.

Consider a case in which there are two hypotheses of interest:

$$H_1 : \mathcal{D} \sim p\left(\mathbf{X}_{1:T}|F^1, \Theta^1\right) \tag{3.24}$$

$$H_2 : \mathcal{D} \sim p\left(\mathbf{X}_{1:T}|F^2, \Theta^2\right). \tag{3.25}$$

Hypothesis H_1 states that a set of time-series are drawn from a FactM with structure F^1 and parameters Θ^1. Similarly H_2 has a structure F^2 and parameters Θ^2. Assuming $p\left(H_1\right) = p\left(H_2\right)$, A binary hypothesis test takes the form:

$$p\left(\mathcal{D}|F^1, \Theta^1\right) \underset{H_2}{\overset{H_1}{\gtrless}} p\left(\mathcal{D}|F^2, \Theta^2\right) \tag{3.26}$$

$$l_{1,2} \triangleq \log \frac{p\left(\mathcal{D}|F^1, \Theta^1\right)}{p\left(\mathcal{D}|F^2, \Theta^2\right)} \underset{H_2}{\overset{H_1}{\gtrless}} 0 \tag{3.27}$$

It is easy to see how this is equivalent to ML over the set of the two possible models:

$$\max_{h \in \{1,2\}} p\left(\mathcal{D}|F^h, \Theta^h\right) \tag{3.28}$$

When analyzing how these hypotheses distinguish themselves from each other it will be useful to define a special factorization, $F^{\cap}$, which describes the common sets of variables which are dependent under both H_1 and H_2. It can be formed by keeping all the unique non-empty intersection sets $F_i^1 \cap F_j^2$ for all $i \in \{1, \ldots, |F^1|\}$ and $j \in \{1, \ldots, |F^2|\}$. For example, for $N = 4$ and

$$F^1 = \{\{1, 2, 3\}, \{4\}\} \tag{3.29}$$

$$F^2 = \{\{1, 2, 4\}, \{3\}\} \tag{3.30}$$

the common factorization is

$$F^{\cap} = \{\{1, 2\}, \{3\}, \{4\}\}. \tag{3.31}$$

Using this common factorization $F^{\cap}$ we can define two conditional distributions using the parameters associated with each hypothesis and the structure $F^{\cap}$: $p\left(\mathbf{X}_t|\widetilde{\mathbf{X}}_t, F^{\cap}, \Theta^1\right)$ and $p\left(\mathbf{X}_t|\widetilde{\mathbf{X}}_t, F^{\cap}, \Theta^2\right)$. Specifically, for $h \in \{1, 2\}$ we define:

$$p\left(\mathbf{X}_t|\widetilde{\mathbf{X}}_t, F^{\cap}, \Theta^h\right) = \prod_{f \in F^{\cap}} p\left(\mathbf{x}_t^f|\widetilde{\mathbf{x}}_t^f, \Theta^h\right) \tag{3.32}$$

where

$$p\left(\mathbf{x}_t^f|\widetilde{\mathbf{x}}_t^f, \Theta^h\right) = \frac{\int p\left(\mathbf{X}_t, \widetilde{\mathbf{X}}_t|F^h, \Theta^h\right) d\mathbf{x}_t^{\mathbf{R}} d\widetilde{\mathbf{x}}_t^{\mathbf{R}}}{\int p\left(\mathbf{X}_t|F^h, \Theta^h\right) d\mathbf{x}_t^{\mathbf{R}}} \tag{3.33}$$

and $\mathbf{R} = \{1, \ldots, N\} \setminus f$ is set of elements not in factor f. That is, $p\left(\mathbf{X}_t|\widetilde{\mathbf{X}}_t, F^{\cap}, \Theta^h\right)$ factorizes according to the common structure $F^{\cap}$ with each factor distribution marginally consistent with hypothesis H_h at time t.

If both the parameters and structure are known the log likelihood ratio for H_1 and H_2 given data $\mathcal{D}$ takes the form :

$$l_{1,2} = \log \frac{p\left(\mathcal{D}|F^1, \Theta^1\right)}{p\left(\mathcal{D}|F^2, \Theta^2\right)} \tag{3.34}$$

$$= \sum_{t=1}^{T} \log \frac{p\left(\mathcal{D}_t|\widetilde{\mathcal{D}}_t, F^1, \Theta^1\right)}{p\left(\mathcal{D}_t|\widetilde{\mathcal{D}}_t, F^2, \Theta^2\right)} \tag{3.35}$$

$$= \sum_{t=1}^{T} \log \frac{p\left(\mathcal{D}_t|\widetilde{\mathcal{D}}_t, F^1, \Theta^1\right)}{p\left(\mathcal{D}_t|\widetilde{\mathcal{D}}_t, F^{\cap}, \Theta^h\right)} \frac{p\left(\mathcal{D}_t|\widetilde{\mathcal{D}}_t, F^{\cap}, \Theta^h\right)}{p\left(\mathcal{D}_t|\widetilde{\mathcal{D}}_t, F^2, \Theta^2\right)} \tag{3.36}$$

where the last step is simply a multiplication by one and h is either 1 or 2. Given H_1 is true, the expectation of the likelihood ratio for a finite realization is:

$$\mathbb{E}_{\mathcal{D}}\left[l_{1,2}|H_1\right] = \sum_{t=1}^{T} D\left(\; p\left(\mathcal{D}_t|\widetilde{\mathcal{D}}_t, F^1, \Theta^1\right) \; \middle\| \; p\left(\mathcal{D}_t|\widetilde{\mathcal{D}}_t, F^{\cap}, \Theta^1\right) \; \right) \tag{3.37}$$

$$+ \sum_{t=1}^{T} D\left(\; p\left(\mathcal{D}_t|\widetilde{\mathcal{D}}_t, F^{\cap}, \Theta^1\right) \; \middle\| \; p\left(\mathcal{D}_t|\widetilde{\mathcal{D}}_t, F^2, \Theta^2\right) \; \right) \tag{3.38}$$

and similarly under H_2,

$$\mathbb{E}_{\mathcal{D}}\left[l_{1,2}|H_2\right] = -\sum_{t=1}^{T} D\left(\; p\left(\mathcal{D}_t|\widetilde{\mathcal{D}}_t, F^2, \Theta^2\right) \; \middle\| \; p\left(\mathcal{D}_t|\widetilde{\mathcal{D}}_t, F^{\cap}, \Theta^2\right) \; \right) \tag{3.39}$$

$$- \sum_{t=1}^{T} D\left(\; p\left(\mathcal{D}_t|\widetilde{\mathcal{D}}_t, F^{\cap}, \Theta^2\right) \; \middle\| \; p\left(\mathcal{D}_t|\widetilde{\mathcal{D}}_t, F^1, \Theta^1\right) \; \right) \tag{3.40}$$

A full derivation is provided in Appendix B.1. Under each hypothesis the expected log likelihood ratio can be decomposed into two sets of KL divergence terms. The first set of terms compares the true structure to the common structure under a consistent set of parameters. They capture the differences in statistical dependence between the hypotheses. The second set of terms compare differences in in both structure and parameters.

As described above, when the true parameters are unknown an ML approach estimates them from the data. This corresponds to a generalized likelihood ratio test (GLRT) in which point estimates of parameters $\hat{\Theta}^1$ and $\hat{\Theta}^2$ are used in place of the true unknown parameters. A GLRT takes the form

$$\hat{l}_{1,2} \triangleq \log \frac{p\left(\mathcal{D}|F^1, \hat{\Theta}^1\right)}{p\left(\mathcal{D}|F^2, \hat{\Theta}^2\right)} \underset{H_2}{\overset{H_1}{\gtrless}} 0. \tag{3.41}$$

Given a single data set $\mathcal{D}$ to analyze, the parameters Θ^1 and Θ^2 are both estimated by maximizing the likelihood of **same** data, under different factorizations. This data came from a single unknown hypothesis. The estimate of the parameters for the true hypothesis will be asymptotically accurate given enough data and a consistent ML estimator. However, the parameter estimates for the incorrect hypothesis will not. Given a consistent estimator (and some assumptions of ergodicity and stationarity) the estimated distribution for the incorrect hypothesis will converge to a FactM with the common structure $F^\cap$ and parameters consistent with the true hypothesis. That is, if the data came from H_1,

$$p\left(\mathcal{D}_t|\widetilde{\mathcal{D}}_t, F^1, \hat{\Theta}^1\right) \overset{H_1}{\to} p\left(\mathcal{D}_t|\widetilde{\mathcal{D}}_t, F^1, \Theta^1\right) \tag{3.42}$$

$$p\left(\mathcal{D}_t|\widetilde{\mathcal{D}}_t, F^2, \hat{\Theta}^2\right) \overset{H_1}{\to} p\left(\mathcal{D}_t|\widetilde{\mathcal{D}}_t, F^\cap, \Theta^1\right) \tag{3.43}$$

and if the data came from H_2

$$p\left(\mathcal{D}_t|\widetilde{\mathcal{D}}_t, F^1, \hat{\Theta}^1\right) \overset{H_2}{\to} p\left(\mathcal{D}_t|\widetilde{\mathcal{D}}_t, F^\cap, \Theta^2\right) \tag{3.44}$$

$$p\left(\mathcal{D}_t|\widetilde{\mathcal{D}}_t, F^2, \hat{\Theta}^2\right) \overset{H_2}{\to} p\left(\mathcal{D}_t|\widetilde{\mathcal{D}}_t, F^2, \Theta^2\right) \tag{3.45}$$

See Appendix B.2 for a full derivation and assumed conditions. Asymptotically the

expected log likelihood ratio of the GLRT , $\hat{l}_{1,2}$, becomes:

$$\mathbb{E}_{\mathcal{D}}\left[\hat{l}_{1,2}|H_1\right] \rightarrow \sum_{t=1}^{T} D\left(p\left(\mathcal{D}_t|\widetilde{\mathcal{D}}_t, F^1, \Theta^1\right) \middle\| p\left(\mathcal{D}_t|\widetilde{\mathcal{D}}_t, F^{\cap}, \Theta^1\right) \right) + 0 \tag{3.46}$$

$$\mathbb{E}_{\mathcal{D}}\left[\hat{l}_{1,2}|H_2\right] \rightarrow -\sum_{t=1}^{T} D\left(p\left(\mathcal{D}_t|\widetilde{\mathcal{D}}_t, F^2, \Theta^2\right) \middle\| p\left(\mathcal{D}_t|\widetilde{\mathcal{D}}_t, F^{\cap}, \Theta^2\right) \right) + 0 \tag{3.47}$$

The terms comparing both parametric and structural differences go to zero and all that is left is the structural comparison under consistent parameters estimated from the data. In other words, the consequence of estimating parameters for the two different models from the same data is the loss of ability to exploit parametric differences.

Nested Hypotheses

Another issue arises when the structures that are being reasoned over are increasingly expressive, yielding nested hypotheses. That is, problems arise when data generated from a FactM using structure F^A can be explained equally well by another which uses a more expressive structure F^B and a specific choice of parameters. A FactM using F^B is more expressive than one using F^A if all factors in F^A are subsets of factors in F^B. Another definition is that F^B is more expressive than F^A if their common structure $F^{\cap} = F^A$. For example, all factorizations are more expressive than the fully independent factorization.

As a consequence of using ML estimation, the more expressive model can always achieve an equal or higher likelihood compared to those less expressive. This will result in always choosing the most expressive model (or the best among a set of most expressive models). Consider only two factorizations $F^1 = \{\{1, 2\}\}$ and $F^2 = \{\{1\}, \{2\}\}$. An ML approach would always choose F^1 when estimating the parameters from a single finite realization of data and $\hat{l}_{1,2}$ will be greater than or equal to zero.

This is a common problem when reasoning over nested models using GLRTs. One standard solution is to estimate significance. That is, estimate a p-value, which says how likely it is that $\hat{l}_{1,2}$ is greater than or equal to the value obtained when data comes from the less expressive model. The p-value is an estimate of probability of false alarm if the more expressive model is chosen. It can be approximated by bootstrap sampling new sets of data from $\mathcal{D}$ which are forced to obey the restrictions of the less expressive model (c.f. [38, 77, 42, 47, 83]). A decision can be made by choosing a threshold. If the p-value is below a set significance value threshold the less expressive hypothesis is

rejected.

Illustrative Example: Dependent versus Independent Gaussians

In order to help understand the above analysis, here, we look at a simple example in which $N = 2$, $r = 0$ and each factor distribution is Gaussian[2]. In this case, the time-series have no temporal dependence and there are only two possible factorizations. Consider the following two hypotheses:

$$H_1 : \mathcal{D}_t \sim p\left(\mathbf{X}_t|F^1,\Theta^1\right) = p\left(\mathbf{x}_t^1,\mathbf{x}_t^2|\Theta_{1,2}^1\right) \tag{3.48}$$
$$= \mathcal{N}\left(\begin{bmatrix}\mathbf{x}_t^1\\ \mathbf{x}_t^2\end{bmatrix};\begin{bmatrix}0\\0\end{bmatrix},\begin{bmatrix}1 & \rho\\ \rho & 1\end{bmatrix}\right)$$
$$H_2 : \mathcal{D}_t \sim p\left(\mathbf{X}_t|F^2,\Theta^2\right) = p\left(\mathbf{x}_t^1|\Theta_1^2\right)p\left(\mathbf{x}_t^2|\Theta_2^2\right) \tag{3.49}$$
$$= \mathcal{N}\left(\mathbf{x}_t^1;\Delta,1\right)\mathcal{N}\left(\mathbf{x}_t^1;\Delta,1\right) = \mathcal{N}\left(\begin{bmatrix}\mathbf{x}_t^1\\ \mathbf{x}_t^2\end{bmatrix};\begin{bmatrix}\Delta\\\Delta\end{bmatrix},\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\right)$$

Hypothesis H_1 corresponds to a dependent factorization with zero mean and dependence ρ, while H_2 assumes independence with mean offset by Δ. Note that the common structure $F^{\cap} = \{\{1\},\{2\}\} = F^2$ due to the fact that F^1 is more expressive than F^2. Furthermore,

$$p\left(\mathbf{X}_t|F^{\cap},\Theta^1\right) = \mathcal{N}\left(\begin{bmatrix}\mathbf{x}_t^1\\ \mathbf{x}_t^2\end{bmatrix};\begin{bmatrix}0\\0\end{bmatrix},\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\right) \tag{3.50}$$
$$p\left(\mathbf{X}_t|F^{\cap},\Theta^2\right) = \mathcal{N}\left(\begin{bmatrix}\mathbf{x}_t^1\\ \mathbf{x}_t^2\end{bmatrix};\begin{bmatrix}\Delta\\\Delta\end{bmatrix},\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\right) = p\left(\mathbf{X}_t|F^2,\Theta^2\right) \tag{3.51}$$

as defined in Equation 3.33.

Since we are dealing with Gaussian distributions, in this case, the expected log likelihood ratio in Equations 3.38 and 3.40 can be computed in closed form. That is,

$$\mathbb{E}_{\mathcal{D}}\left[l_{1,2}|H_1\right] = T\left(-\frac{1}{2}\log(1-\rho^2)\right) \quad +T\left(2\Delta^2\right) \tag{3.52}$$
$$\mathbb{E}_{\mathcal{D}}\left[l_{1,2}|H_2\right] = 0 \quad -T\left(\frac{1}{2}\log(1-\rho^2) + 2\frac{1+\Delta^2(1-\rho)}{1-\rho^2} - 2\right) \tag{3.53}$$

[2]This is similar to the example presented in [47]. Here, the parametric differences are in terms of the mean rather than marginal variance.

These expectations are true when the parameters associated with each hypothesis are known. When the parameters are unknown, an ML approach estimates the parameters for each factorization from the data. In this case, one would obtain an ML estimate of the mean and covariance of the given data $\mathcal{D}$:

$$\hat{\mu} = \begin{bmatrix} \hat{\mu_1} \\ \hat{\mu_2} \end{bmatrix}, \qquad \hat{\Lambda} = \begin{bmatrix} \hat{\sigma}_1^2 & \hat{\rho}\hat{\sigma}_1\hat{\sigma}_2 \\ \hat{\rho}\hat{\sigma}_1\hat{\sigma}_2 & \hat{\sigma}_2^2 \end{bmatrix} \tag{3.54}$$

Using these ML parameter estimates in place of the true parameters we obtain

$$p\left(\mathbf{X}_t|F^1, \hat{\Theta}^1\right) = \mathcal{N}\left(\begin{bmatrix} \mathbf{x}_t^1 \\ \mathbf{x}_t^2 \end{bmatrix}; \hat{\mu}, \begin{bmatrix} \hat{\sigma}_1^2 & \hat{\rho}\hat{\sigma}_1\hat{\sigma}_2 \\ \hat{\rho}\hat{\sigma}_1\hat{\sigma}_2 & \hat{\sigma}_2^2 \end{bmatrix}\right) \tag{3.55}$$

$$p\left(\mathbf{X}_t|F^2, \hat{\Theta}^2\right) = \mathcal{N}\left(\begin{bmatrix} \mathbf{x}_t^1 \\ \mathbf{x}_t^2 \end{bmatrix}; \hat{\mu}, \begin{bmatrix} \hat{\sigma}_1^2 & 0 \\ 0 & \hat{\sigma}_2^2 \end{bmatrix}\right), \tag{3.56}$$

to use when performing maximum likelihood inference. The mean and marginal variance estimates will be the same for each factorization since the parameters are estimated from the same data. Note that the only difference is that in the dependent model we use the estimate of correlation $\hat{\rho}$. $p\left(\mathcal{D}|F^1, \hat{\Theta}^1\right)$ will always be greater than or equal to $p\left(\mathcal{D}|F^2, \hat{\Theta}^2\right)$. Thus, F^1 would always be chosen. This can also be see by examining the expected log likelihood ratios in the equivalent GLRT:

$$\mathbb{E}_{\mathcal{D}}\left[\hat{l}_{1,2}|H_1\right] = T\left(-\frac{1}{2}\log(1-\hat{\rho}^2)\right) \qquad +0 \tag{3.57}$$

$$\mathbb{E}_{\mathcal{D}}\left[\hat{l}_{1,2}|H_1\right] = 0 \qquad -0 \tag{3.58}$$

All that remains are terms involving estimates of dependence. In this case, $\hat{l}_{1,2}$ is simply an estimate of mutual information,$I(\mathbf{x}_t^1; \mathbf{x}_t^2)$.

An analytic form for the p-value can be obtained here since we are dealing with Gaussian distributions[3]. However, in general, when performing a test of dependence for $N = 2$ and $r = 0$ a p-value can be computed by simulating data from the independent factorization. This can be done by simply permuting the time index for $\mathcal{D}^1$. Under the independent hypothesis, all such permutations should be equally likely[4]. For each

[3]The GLRT takes the form a well studied problem of deciding whether ρ is non-zero given the empirical estimate.

[4]If $r > 0$ one can approximate samples from the independent factorization by randomly shifting one time-series relative to the other. Alternatively one could form an augmented sample Y_t which contains $\mathcal{D}_t$ and $\widetilde{\mathcal{D}}_t$, do permutations within each factor of this augmented joint sample, and then extract new sets if D_t and $\widetilde{\mathcal{D}}_t$ to be used in calculating $\hat{l}_{1,2}$.

permutation a different estimate of $\hat{\rho}$ and $\hat{l}_{1,2}$ can be computed. The p-value can be calculated by counting the number of times a $\hat{l}_{1,2}$ exceed the value obtained on the non-permuted data. This p-value can then be used to make the decision of whether to reject the independent hypothesis H_2.

3.4 Temporal Interaction Model

The factorization model presented in the previous section describes multiple time-series in terms of collections of independent groups. It leaves the details of how the time-series within a single group evolve abstract. That is, the details are problem dependent and implicitly defined by choice of parameterization for $p\left(\mathbf{x}_t^{F_f}|\widetilde{\mathbf{x}}_t^{F_f}, \Theta_{F_f}\right)$.

In this section, we introduce an alternative model. A *temporal interaction model*, TIM(r), explicitly describes the details of the causal dependence among time-series using a directed graph $\bar{E}$. Specifically, $\bar{E}$ is a directed structure on N vertices. Each vertex corresponds to a time-series and the directed edges in the graph represent the causal dependence among time series. A TIM(r) model with structure $\bar{E}$ and parameters Θ factorizes as

$$p\left(\mathbf{X}_t|\widetilde{\mathbf{X}}_t, \bar{E}, \Theta\right) = \prod_{v=1}^{N} p\left(\mathbf{x}_t^v|\widetilde{\mathbf{x}}_t^{v,\mathbf{pa}(v)}, \Theta_{v|\mathbf{pa}(v)}\right), \tag{3.59}$$

where, as defined in Chapter 2, $\mathbf{pa}\,(v)$ returns the parents of vertex v given the structure $\bar{E}$. The v-th time-series at time t is dependent on its own past $\widetilde{\mathbf{x}}_t^v$ as well as the past of the time-series in its parent set $\mathbf{S} = \mathbf{pa}\,(v)$, $\widetilde{\mathbf{x}}_t^{\mathbf{S}}$. The parameters $\Theta_{v|\mathbf{S}}$ describe the nature this dependence[5].

Figure 3.3(a) illustrates a TIM(1) as a directed Baysian network for $N = 3$ time-series. Here, $\bar{E}$ containing two edges; one from 2 to to 1, and one from 2 to 3 and thus $\mathbf{pa}\,(1) = \mathbf{pa}\,(3) = 2$ and $\mathbf{pa}\,(2) = \emptyset$. Note that $\bar{E}$ only describes the edges across time-series and the edges representing temporal dependence on past values of the same time-series are a result of our base Markov assumption in static dependence models. Figure 3.3(b) shows the alternative dependence graph view of this model which hides within time-series dependence by representing each time-series as a single vertex. We

[5]By definition, in a TIM(r), each time-series at time t is always dependent on its own past. Thus, we use $\Theta_{v|\mathbf{S}}$ rather than the more explicit notation $\Theta_{v|v,\mathbf{S}}$ to represent the parameters of this relationship for brevity. One can image alternative models in which this is not the case, but we leave such models for future work.

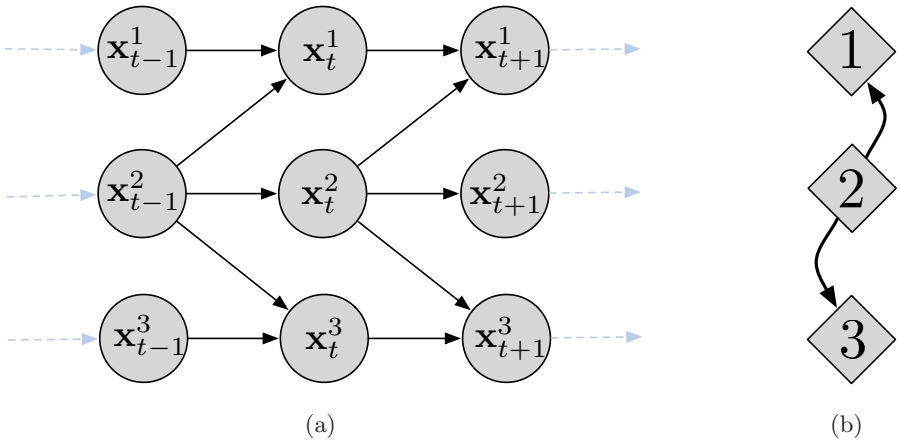


Figure 3.3. *Example TIM(1) Model Structure and Interaction Graph:* The directed structure of a TIM(1) for $N = 3$ time-series is shown in (a) and the corresponding interaction graph is shown in (b).

will often refer to this form of dependence graph as an *interaction graph* in that it details the interactions among time-series. There is a one to one mapping between the Bayesian network for a TIM(r) and its interaction graph. A directed edge from u to v in the interaction graph implies a directed edge from $\widetilde{\mathbf{x}}^u_t$ to $\mathbf{x}^v_t$ in the TIM(r). As mentioned previously, this interaction graph is not meant to be interpreted as a graphical model. It simply provides a representation in which the structure $\bar{E}$ can be easily read from.

Note that the TIM(1) in Figure 3.3(a) has an interaction graph that happens to be fully connected and acyclic. However, in general it need not be. In other words, the space of interaction graphs is the set of all directed graphs. Cycles in the interaction graph do not result in cycles in the Bayesian network for the TIM(r). This is due to the assumption of temporal causality. The edges specified in $\bar{E}$ specifically describe causal dependence from past values of time-series to current values. We show in the Sections 3.4.2 and 3.4.3 that this causal assumption and lack of constraints on $\bar{E}$ will allow for efficient reasoning over structure.

It is important to understand the differences between a TIM(r) and the previously introduced FactM(r). A TIM(r) assumes that the time-series values at the t are conditionally independent of each other given the past values of their parents. This assumption is not made in a FactM(r). This begs the question, why not use a FactM(r) which has fewer restrictive assumptions? While the FactM(r) has more modeling power, it

achieves this by leaving its parameterization more abstract. The structure F in the FactM only explicitly indicates independent groups of time-series. It leaves the details on how current time-series within a group depend on their past to be implicitly defined by the chosen parameterization and parameter values.

A TIM(r) explicitly describes these causal relationships via $\bar{E}$. More can be understood about the statistical dependence relationships from obtaining an estimate of structure rather than looking at the particular parameterization or values of the parameters. Take for example the TIM(1) shown in Figure 3.3(a). A FactM(1) with structure $F = \{\{1, 2, 3\}\}$ and an appropriate choice for $\Theta_{1,2,3}$ can equivalently model data drawn from this TIM(1). However, very little information is given by structure F. In fact, data from any TIM(1) that uses a fully connected directed structure can be equivalently modeled with a FactM(r) whose structure $F = \{\{1, \ldots, N\}\}$ and the appropriate choice of $\Theta_{1,\ldots,N}$. In other words, the directed structure used by a TIM(r) imposes more constraints on the dependence relationships among time-series.

Once these differences are understood, the choice of model is dependent on which best matches the end application of interest. We will use a FactM for reasoning over audio-visual associations in Chapter 5 and a TIM for understanding interactions among moving objects in Chapter 6. In our audio-visual association task we are simply interested in whether not there is any dependence among an audio and video stream, while in the object interaction analysis task our goal is to characterize the finer details of the causal relationships among multiple moving objects.

■ 3.4.1 Sets of Directed Structures

The structure of a TIM(r) is in the form of a directed graph on N vertices, with edges specified by $\bar{E}$. In this section, we explore how the size of the set of possible structures grows as a function N. We begin by characterizing the set of all possible directed structures.

Definition 3.4.1 ($\mathcal{A}_N$): *$\mathcal{A}_N$ is the set of all directed graphs on N vertices. These graphs are allowed to have cycles but no self-loops, $(v \rightarrow v)$.*

For $\bar{E} \in \mathcal{A}_N$ there are 2^{N-1} possible parent sets for each vertex in the graph. This

yields a super-exponential number of possible directed structures:

$$|\mathcal{A}_N| = \left(2^{N-1}\right)^N = 2^{N^2-N}. \tag{3.60}$$

It is simple to see from inspection that the set of all structures for a TIM(r) is larger than that of a FactM(r), $|\mathcal{A}_N| \geq |\mathcal{B}_N|$. Again, this is why a TIM can explicitly specify more detailed dependence relationships in its structure. In the following sub sections we will discuss increasingly restrictive subsets of $\mathcal{A}_N$.

Bounded Parent Set

It may be desirable to limit the types of directed structures considered for a TIM(r). Constraining the set of structures can allow for more efficient modeling and inference. One simple constraint is to limit the number of parents any vertex has in the graph specified by $\bar{E}$. That is, one can limit the in-degree of each vertex.

Definition 3.4.2 ($\mathcal{P}_N^K$): *$\mathcal{P}_N^K \subset \mathcal{A}_N$ is the set of all directed structures on N vertices in which each vertex has no more than K parents.*

A TIM(r) with structure $\bar{E} \in \mathcal{P}_N^K$ constraints the maximum number of "influences" for each time-series. Given N verticies, each has

$$\sum_{k=1}^{K} \binom{N-1}{k} \leq N^K \tag{3.61}$$

possible parent sets of a size less than or equal to K. While this reduces the number of possible structures, there is still a super-exponential number of them. That is $|\mathcal{P}_N^K|$ is $O(N^{NK})$.

Directed Trees and Forests

The set $\mathcal{P}_N^K$ imposes a local constraint on parent set size. However, there may be situations in which a global structure constraint is desirable. For example, one may want to only consider directed structures which are acyclic and connected.

Definition 3.4.3 ($\mathcal{T}_N$): *$\mathcal{T}_N \subset \mathcal{P}_N^1$ is the set of all directed tree structures on N vertices. A directed tree (rooted spanning arborescence) is a fully connected structure in which $N-1$ vertices have a single parent while the remaining vertex has no parents and is designated as the root.*

If a TIM(r) has a structure $\bar{E} \in \mathcal{T}_N$ there exists a root time-series which has influence on all others given enough time due to connectivity. In addition, each time-series can never be influenced by one of its children or any time-series influenced by its children.

There are $|\mathcal{T}_N| = N^{N-1}$ directed trees on N vertices. A simple proof comes from first considering the space of all undirected trees. Cayley's formula states that for any integer N, the number of undirected trees on N labeled vertices is N^{N-2} [16]. An undirected tree can always be converted to a directed tree by picking a root and then directing all edges away from the root in succession. Since there are N ways to choose the root there are $N(N^{N-2}) = N^{N-1}$ directed trees.

Definition 3.4.4 ($\mathcal{F}_N$): *$\mathcal{F}_N \subset \mathcal{P}_N^1$ is the set of all directed forest structures on N vertices. A directed forest is an acyclic graph with each vertex having at most one parent. A directed forest can have multiple roots.*

The set $\mathcal{F}_N$ removes the fully connected assumption used to define $\mathcal{T}_N$. There are $(N+1)^{(N-1)}$ directed forests on N vertices. A simple proof, again, uses a mapping from undirected trees. Given an undirected tree on $N+1$ vertices, one can form a directed forest on N vertices by choosing a special super-root vertex, creating a directed tree outward from this super-root and then removing that vertex and all its outgoing edges. The roots of each tree in the directed forest are the children of the super-root vertex in the undirected tree. Thus, there are $|\mathcal{F}_N| = (N+1)^{(N+1-2)} = (N+1)^{N-1}$ possible directed forests.

Note that $\mathcal{T}_N \subset \mathcal{F}_N \subset \mathcal{P}_N^1$ and all grow super-exponentially with N. Figure 3.4 plots the number of structures for the various sets discussed in this section as a function of N. We will use TIMs in scenarios in which N may be large and the allowable dependence relationships are not specified by the problem of interest. That is, we wish to reason over these large sets of directed structures. Specifically in Chapter 6 we will reasoning over as many as 11^{10} structures of interactions among moving objects[6].

■ 3.4.2 Prior on TIM Parameters and Structure

We will focus on performing exact Bayesian inference as described in Section 3.2.1 when using a TIM. In this section, we present prior models for the parameters and structures,

[6] We will consider directed tree structures when describing the relationships among 10 players and a ball in a basketball game

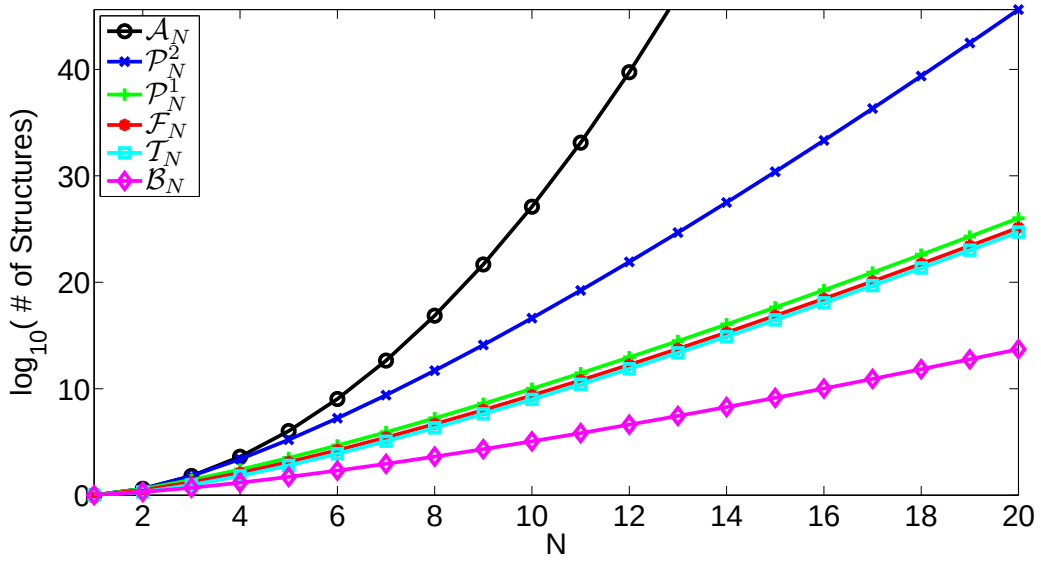


Figure 3.4. *The Size of Sets of Directed Structure vs N:* The number of possible structures as a function of N for the set of all directed structures $\mathcal{A}_N$, directed structures with at most 2, $\mathcal{P}_N^2$, and 1, $\mathcal{P}_N^1$ parents, directed forests $\mathcal{F}_N$, directed trees $\mathcal{T}_N$, and factorizations $\mathcal{B}_N$.

allowing one to specify prior knowledge about these unobserved variables. We adopt a prior on the structure and parameters similar to those presented in [62, 33], using the factorization

$$p_0\left(E,\Theta\right) = p_0\left(E\right)p_0\left(\Theta|E\right). \tag{3.62}$$

Again, there are various challenges in designing these priors. The first challenge is due to the fact that the number of the parameters specified by Θ (and potentially the values) is a function of the specific structure. For example, in a TIM(r) the number of parameters is a function of the number of edges in $\bar{E}$. A second challenge was alluded to in Section 3.4.1. That is, for TIMs there are a super-exponential number of structures to consider. The question remains on how one places a tractable prior over this large space of structures.

We begin by discussing the prior on parameters given a specified structure, $p_0\left(\Theta|\bar{E}\right)$. We adopt a form for this prior based on two key properties:

1. The distribution factorizes according to the structure and parameters are assumed to be independent of each other.

2. The parameters are *modular*. That is, the prior on parameters associated with the v-th time-series given its parent set $\mathbf{pa}(v)$ is not a function of the global structure describing the parent sets of all other time-series.

Given a structure $\bar{E}$ and hyperparameters Υ we assume the prior on parameters factorizes according to the edge structure such that

$$p_0\left(\Theta|\bar{E}\right) = \prod_{v=1}^{N} p_0\left(\Theta_{v|\mathbf{pa}(v)}|\Upsilon\right). \tag{3.63}$$

Here, $p_0\left(\Theta_{v|\mathbf{S}}|\Upsilon\right)$ is modular. That is, it is the same for all structures $\bar{E}$ for which $\mathbf{S}$ is the parent set of v. Thus, for each time-series one needs to specify a parameter prior for all potential parent sets. Given this finite set of priors for each v one is able to construct a full prior on parameters for any given structure $\bar{E}$ using Equation 3.63.

As discussed in Section 3.4.1 there is an exponential number, 2^{N-1}, of possible parent sets when $\bar{E} \in \mathcal{A}_N$. However, by bounding the number of parents to K such that $\bar{E} \in \mathcal{P}_N^K$, one only needs to specify a polynomial number of prior terms.

Next we focus on specifying the structural prior, $p_0\left(\bar{E}\right)$. One desirable property we wish to obtain is the ability to favor certain structures over others. Another is that the prior lends itself to tractable inference. Tractability will be linked to computation of the partition function. We adopt a structural prior which allows one to favor certain parent sets over others. While this prior is over a set of structures which grows super-exponentially with N, we show that it can be reasoned over in exponential-time. Furthermore, we show that by bounding the size of parents sets in $\bar{E}$ one can tractably calculate the partition function in polynomial-time.

The structural prior we adopt has the form

$$p_0\left(\bar{E}\right) = \frac{1}{Z(\beta)} \prod_{v=1}^{N} \beta_{\mathbf{pa}(v),v}, \tag{3.64}$$

where, $Z(\beta)$ is the partition function. Each scalar hyperparameter $\beta_{\mathbf{S},v}$ can be interpreted as a weight on the parent set $\mathbf{S}$ for v-th time-series. The prior is simply proportional to the product of these weights. Note that if all $\beta_{\mathbf{S},v}$ are set to 1, one obtains a uniform prior and $Z(\beta)$ is the number of possible structures. If all $\beta_{\mathbf{S},v}$ are set proportional to the size of $\mathbf{S}$, $|\mathbf{S}|$, the prior will favor dense structures. Equivalently, sparse structures are favored by making $\beta_{\mathbf{S},v}$ inversely proportional to $|\mathbf{S}|$.

As noted in Section 3.4.1 there are 2^{N^2-N} possible structures $\bar{E} \in \mathcal{A}_N$. This suggests that one may need to explicitly sum over a super exponential number of terms when

calculating $Z(\beta)$. Fortunately, however, when $\bar{E} \in \mathcal{A}_N$, all combinations of parent sets are allowable and there are no global structural constraints limiting the parent set of one vertex given another. Thus, one can calculate the partition function,

$$\begin{aligned} Z(\beta) &= \sum_{\bar{E}\in\mathcal{A}} \prod_{v=1}^{N} \beta_{\mathbf{pa}(v),v} \\ &= \sum_{\mathbf{S}_1} \cdots \sum_{\mathbf{S}_N} \prod_{v=1}^{N} \beta_{\mathbf{S}_v,v} \\ &= \prod_{v=1}^{N} \sum_{\mathbf{S}_v} \beta_{\mathbf{S}_v,v} \\ &\triangleq \prod_{v=1}^{N} \gamma_v(\beta), \end{aligned} \tag{3.65}$$

as a product of N summations where a sum over $\mathbf{S}_v$ is simply the sum over all 2^{N-1} allowable parent sets for v. Each $\gamma_v(\beta)$ is defined to be this summation for a particular v. The switch of the product and sums in the above equation is possible due to fact that parent set for one vertex is independent of all others. This allows $Z(\beta)$ to be calculated a product of N summations, each of which contains 2^{N-1} terms. Thus, one can reason over all structures in exponential time, $N2^{N-1}$, rather than super-exponential time. While this is a large improvement, the exponential number of terms quickly becomes intractable for large N. In the next sections we show how using the simple constraints discussed in Section 3.4.1 one can obtain $Z(\beta)$ in polynomial time while still considering a super-exponential number of candidate structures.

Bounded Parent Sets

Bounding the size of the parent set for each time-series, $\bar{E} \in \mathcal{P}_M^K$, yields a partition function that has the same form as Equation 3.65 with

$$\gamma_v(\beta) = \sum_{\mathbf{S}_v,\ s.t.\ \ |\mathbf{S}_v|\leq K} \beta_{\mathbf{S}_v,v}. \tag{3.66}$$

This is again due to the fact that the constraint imposed $\mathcal{P}_M^K$ is local and the parent set of each vertex (time-series) can be treated independently of all others. The partition becomes a sum of all possible parent sets of size less than or equal to K. One can bound the order of the summation by $\sum_{k=1}^{K} \binom{N-1}{k} \leq N^K$. Thus, only polynomial

computation time, $O(N^{K+1})$, is needed to calculate $Z(\beta)$ even though the total number of structures is still super exponential, $O(N^{NK})$.

Directed Trees and Forests

If $\bar{E} \in \mathcal{T}_N$, it is restricted by both local and global constraints. The local constraint ensures each vertex has at most one parent, while the global constraints restrict the structure to be acyclic and fully connected. Due to the local single parent constraint, in this context, β can be thought of as an $N \times N$ matrix with each hyperparameter $\beta_{u,v\neq u}$ being interpreted as a weight on the edge $u \to v$, and $\beta_{\emptyset,v} \triangleq \beta_{v,v}$ as a weight on a vertex being a root. The edge set corresponding to the nonzero entries of β form a support graph. We will assume this support graph is connected and contains at least one directed tree.

While there are N^{N-1} possible directed trees on N vertices, the Matrix Tree Theorem allows one to calculate $Z(\beta)$ in polynomial time. This theorem was used by Meila and Jaakkola [62] for reasoning over undirected trees. The undirected version of theorem is a special case of the often rediscovered, real-valued, directed version, originally developed by Kirchhoff [51] in 1847. The theorem allows one to calculate the weighted sum over all directed trees rooted at r, $Z_r(\beta)$ via

$$Z_r(\beta) = \sum_{\bar{E} \text{ rooted at } r} \prod_{u \to v} \beta_{u,v} = Cof_{r,r}\left(\bar{Q}(\beta)\right), \tag{3.67}$$

where $\bar{Q}(\beta)$ is the Kirchhoff matrix with its u, v entry defined as

$$\bar{Q}_{u,v}(\beta) = \begin{cases} -\beta_{u,v} & 1 \leq u \neq v \leq N \\ \sum_{u'=1}^{N} \beta_{u',v} - \beta_{u,u} & 1 \leq u = v \leq N \end{cases} \tag{3.68}$$

and $Cof_{i,j}(M)$ is the i, j cofactor of matrix M. $Cof_{i,r}\left(\bar{Q}(\beta)\right)$ is invariant to i and gives the sum over all weighted trees rooted at r. A proof can be found in [90].

By summing over all N possible roots one obtains

$$Z(\beta) = \sum_{r=1}^{N} \beta_{r,r} Z_r(\beta). \tag{3.69}$$

Thus, a straightforward implementation yields $O(N^4)$ time for calculating the partition function: $O(N^3)$ for each of the N $Z_r(\beta)$ terms. However, as pointed out by Koo et. al [56], a useful observation allows for only $O(N^3)$ time computation of $Z(\beta)$. That is, it

can be calculated via a single determinant,

$$Z(\beta) = |\hat{Q}(\beta)|, \tag{3.70}$$

where

$$\hat{Q}_{u,v}(\beta) = \begin{cases} \beta_{u,u} & u = 1 \\ \bar{Q}_{u,v}(\beta) & u > 1 \end{cases} \tag{3.71}$$

The proof follows from the construction of $\hat{Q}_{u,v}$ and the invariance of $Cof_{i,r}\left(\bar{Q}(\beta)\right)$ to i:

$$|\hat{Q}(\beta)| = \sum_{v=1}^{N} \hat{Q}_{1,v}(\beta)\, Cof_{1,v}\left(\hat{Q}(\beta)\right) \tag{3.72}$$

$$= \sum_{v=1}^{N} \beta_{v,v}\, Cof_{1,v}\left(\bar{Q}(\beta)\right) \tag{3.73}$$

$$= \sum_{v=1}^{N} \beta_{v,v} Z_v(\beta) \tag{3.74}$$

$$= Z(\beta) \tag{3.75}$$

Consider a simple case in which $N = 2$ and all βs are set to 1. In this case $Z(\beta)$ is a count of the number of trees on two vertices. The matrix $\bar{Q}(\beta) = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$, $Z_1(\beta) = Z_2(\beta) = 1$ and thus $Z(\beta) = 2$ which is the number of possible directed trees on two vertices.

Directed forests, $\bar{E} \in \mathcal{F}_N$, remove the fully connected assumption of $\mathcal{T}_N$ and can have multiple roots. While this yields a larger set of structures, $Z(\beta)$ can still be calculated in $O(N^3)$ time, as show in [56]. Specifically,

$$Z(\beta) = |\bar{Q}(\beta) + \text{diag}(\beta_{1,1}, \ldots, \beta_{N,N})| \tag{3.76}$$

where $\text{diag}(a_1, ..., a_N)$ is a diagonal matrix with the vector $\mathbf{a}$ on its diagonal. This is a consequence of that fact that any directed forest can be turned into a directed tree by the addition of one virtual super-root vertex which has no parents and connects to all the roots of the trees within the forest. Each $\beta_{v,v}$ can be interpreted as a weight on an outgoing edge from this super-root vertex to vertex v.

Again, consider the case when $N = 2$ and all βs are set to 1. Here, $Z(\beta) = \left|\begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}\right| = 3$, corresponding to the fact there are three possible forests.

Note that while $\mathcal{T}_N \subset \mathcal{F}_N \subset \mathcal{P}_N^1$, both directed trees and forests require $O(N^3)$ computation even though the larger class of $\mathcal{P}_N^1$ only requires $O(N^2)$. This is due to the imposed global acyclic constraint which limits the parent set of one vertex based on the parent sets of others.

■ 3.4.3 Bayesian Inference of TIM Structure

Now that we have described the generative model used by a TIM and placed tractable priors on the parameters and structures, we turn to the task of Bayesian inference over these structures. Specifically we wish to calculate (as presented in Equation 3.4):

$$p\left(\bar{E}|\mathcal{D}\right) = \frac{1}{p\left(\mathcal{D}\right)} p_0\left(\bar{E}\right) p\left(\mathcal{D}|\bar{E}\right)$$

using the priors presented in the previous section. We show that both the prior over structure and the prior on parameters given structure are conjugate, allowing for tractable inference. In addition, given the full posterior over structure we show how to calculate substructure appearance posteriors and expectations. Lastly, we discuss MAP estimates of structure.

Conjugacy

We begin by examining the joint posterior over structure $\bar{E}$ and parameters Θ given data $\mathcal{D}$. Consider the following factorization of the posterior:

$$p\left(\bar{E}, \Theta|\mathcal{D}\right) = p\left(\Theta|\bar{E}, \mathcal{D}\right) p\left(\bar{E}|\mathcal{D}\right). \tag{3.77}$$

Below we examine each posterior term and show that the priors presented in Section 3.4.2 are conjugate. We begin with the posterior over parameters given structure, $p\left(\Theta|\bar{E}, \mathcal{D}\right)$.

Proposition 3.4.1 : *If one chooses priors $p_0\left(\Theta_{v|\boldsymbol{pa}(v)}|\Upsilon\right)$ that are conjugate for their corresponding conditional distributions $p\left(\mathbf{x}_t^v|\mathbf{x}_t^{v,\boldsymbol{pa}(v)}, \Theta_{v|\boldsymbol{pa}(v)}\right)$, then the full prior $p_0\left(\Theta|\bar{E}\right)$ presented in Equation 3.63 is conjugate for a TIM(r).*

Proof. Using the form of prior on parameters given structure specified in Equation 3.63

the posterior is:

$$p\left(\Theta|\bar{E},\mathcal{D}\right) = \frac{p\left(\mathcal{D}|\Theta,\bar{E}\right)p_0\left(\Theta|\bar{E}\right)}{p\left(\mathcal{D}|\bar{E}\right)} \tag{3.78}$$

$$= \prod_{t=1}^{T}\prod_{v=1}^{N} \frac{p\left(\mathcal{D}_t^v|\mathcal{D}_t^{v,\mathbf{pa}(v)},\Theta_{v|\mathbf{pa}\left(v,\bar{E}\right)}\right)p_0\left(\Theta_{v|\mathbf{pa}(v)}|\Upsilon\right)}{p\left(\mathcal{D}_t^v|\widetilde{\mathcal{D}}_t^{v,\mathbf{pa}(v)},\bar{E}\right)} \tag{3.79}$$

$$= \prod_{t=1}^{T}\prod_{v=1}^{N} p\left(\Theta_{v|\mathbf{pa}(v)}|\mathcal{D}_t^v,\widetilde{\mathcal{D}}_t^{v,\mathbf{pa}(v)},\Upsilon\right) \tag{3.80}$$

$$= \prod_{v=1}^{N} p\left(\Theta_{v|\mathbf{pa}(v)}|\mathcal{D}^{v,\mathbf{pa}(v)},\Upsilon\right). \tag{3.81}$$

It takes the same form as the prior. That is, it is independent over each time-series and modular. Each term is the posterior on parameters for the v-th time-series given its parents. Choosing a $p_0\left(\Theta_{v|\mathbf{pa}(v)}|\Upsilon\right)$ which is conjugate results in a fully conjugate prior on parameters given structure. Thus, the posterior can be obtained by a simple update to the hyperparameters Υ for each term in prior using the data $\mathcal{D}$. □

Next, we turn to the posterior on structure $p\left(\bar{E}|\mathcal{D}\right)$, which is our primary quantity of interest.

Proposition 3.4.2 : *The prior $p_0\left(\bar{E}\right)$ presented in Equation 3.64 is conjugate for a TIM(r).*

Proof. We follow a similar derivation to that used in [62] for obtaining the posterior on undirected trees[7]. Using the prior on structure given in Equation 3.64, we start with the following form:

$$p\left(\bar{E}|\mathcal{D}\right) = \frac{1}{p\left(\mathcal{D}\right)}p\left(\mathcal{D}|\bar{E}\right)p_0\left(\bar{E}\right) \tag{3.82}$$

$$= \frac{1}{p\left(\mathcal{D}\right)}p\left(\mathcal{D}|\bar{E}\right)\frac{1}{Z(\beta)}\prod_{v=1}^{N}\beta_{\mathbf{pa}(v),v}. \tag{3.83}$$

[7] That is, while here we are deriving the posterior for a different and much broader class of structures, the steps in the derivation are similar to [62].

Next, we examine the data evidence given structure by integrating out parameters:

$$p\left(\mathcal{D}|\bar{E}\right) = \int p\left(\mathcal{D}|\Theta,\bar{E}\right) p_0\left(\Theta|\bar{E}\right) d\Theta \tag{3.84}$$

$$= \int \left[\prod_{t=1}^{T}\prod_{v=1}^{N} p\left(\mathcal{D}_t^v|\widetilde{\mathcal{D}}_t^{v,\mathbf{pa}(v)},\Theta_{v|\mathbf{pa}\left(v,\bar{E}\right)}\right) p_0\left(\Theta_{v|\mathbf{pa}(v)}|\Upsilon\right)\right] d\Theta \tag{3.85}$$

$$= \prod_{v=1}^{N}\int\prod_{t=1}^{T} p\left(\mathcal{D}_t^v|\widetilde{\mathcal{D}}_t^{v,\mathbf{pa}(v)},\Theta_{v|\mathbf{pa}\left(v,\bar{E}\right)}\right) p_0\left(\Theta_{v|\mathbf{pa}(v)}|\Upsilon\right) d\Theta_{v|\mathbf{pa}(v)} \tag{3.86}$$

$$= \prod_{v=1}^{N} p\left(\mathcal{D}^v|\widetilde{\mathcal{D}}^{v,\mathbf{pa}(v)},\Upsilon\right) \tag{3.87}$$

$$\triangleq \prod_{v=1}^{N} W_{\mathbf{pa}(v),v} \tag{3.88}$$

The data evidence given structure is a product of terms, each of which is the evidence of the v-th time-series given its parents specified by $\bar{E}$. That is, for parent set $\mathbf{S}$ the evidence for the v-th time-series is :

$$\begin{aligned} W_{\mathbf{S},v} &= p\left(\mathcal{D}^v|\widetilde{\mathcal{D}}^{v,\mathbf{S}},\Upsilon\right) \\ &= \int p\left(\mathcal{D}^v|\widetilde{\mathcal{D}}^{v,\mathbf{S}},\Theta_{v|\mathbf{S}}\right) p_0\left(\Theta_{v|\mathbf{S}}|\Upsilon\right) d\Theta_{v|\mathbf{S}} \end{aligned} \tag{3.89}$$

As mentioned in Chapter 2, for continuous observations and a linear gaussian model with parameters $\Theta_{v|\mathbf{S}}$ one can choose $p_0\left(\Theta_{v|\mathbf{S}}|\Upsilon\right)$ to be a matrix-normal-inverse-Wishart distribution with hyperparameters Υ. This will yield efficient updates for Equation 3.81 and $W_{v|\mathbf{S}}$ will be the evaluation of a Matrix-T distribution. For discrete observations, one can use Dirichlet priors and have analytic forms for the evidence. Specific examples will be given in Chapter 6.

Substituting Equation 3.88 into Equation 3.83 one obtains

$$p\left(\bar{E}|\mathcal{D}\right) = \frac{1}{p\left(\mathcal{D}\right)}\frac{1}{Z(\beta)}\prod_{v=1}^{N}\beta_{\mathbf{pa}(v),v}W_{\mathbf{pa}(v),v}. \tag{3.90}$$

Using the fact that the distribution must sum to 1 we solve for the data evidence $p(\mathcal{D})$:

$$1 = \sum_{\bar{E}} p\left(\bar{E}|\mathcal{D}\right) \tag{3.91}$$

$$1 = \frac{1}{p(\mathcal{D})}\frac{1}{Z(\beta)}\sum_{\bar{E}}\prod_{v=1}^{N}\beta_{\mathbf{pa}(v),v}W_{\mathbf{pa}(v),v} \tag{3.92}$$

$$1 = \frac{1}{p(\mathcal{D})}\frac{1}{Z(\beta)}Z(\beta \circ W) \tag{3.93}$$

$$p(\mathcal{D}) = \frac{Z(\beta \circ W)}{Z(\beta)}, \tag{3.94}$$

where $\circ$ is an element wise (Hadamard) product. Each $\beta_{\mathbf{S},v}$ is multiplied by $W_{\mathbf{S},v}$. This yields the posterior,

$$p\left(\bar{E}|\mathcal{D}\right) = \frac{1}{Z(\beta \circ W)}\prod_{v=1}^{N}\beta_{\mathbf{pa}(v),v}W_{\mathbf{pa}(v),v}. \tag{3.95}$$

Thus, the prior is conjugate and the posterior is obtained by updating the hyperparameters β via multiplication of data evidence weights W. □

Structure Event Probabilities and Expectations

The ability to compute the partition function and conjugacy of the prior enables exact computation of a wide variety of useful prior and/or posterior event probabilities. Here, we present some examples in terms of the prior distribution on structure. However, conjugacy allows one to convert these to posterior events by substituting $\beta \circ W$ in place of β.

The probability of a particular edge being present is:

$$p\left(I_{u\to v} = 1\right) = \mathbb{E}\left[I_{u\to v}\right] = 1 - \frac{Z(\beta^{-(u\to v)})}{Z(\beta)} \tag{3.96}$$

where $I_{u\to v}$ is an indicator variable that has value 1 when the edge $u \to v$ is present. $\beta^{-(u\to v)}$ is β with all elements involving edge from u to v set to zero. In words, Equation 3.96 calculates the probability of an edge as 1 minus the probability of that edge not appearing. Interpreting the partition function as a weighted count of possible structures, the probability of an edge not appearing is the ratio of a weight count of structures which do not include that edge to the weighted count of all structures.

Using the same approach one can calculate the joint edge appearance probability of one set of edges conditioned on another set. Similarly, one can calculate the probability

a time-series has no parents (is a root) or no children (is a leaf):

$$p\left(I_v \text{ is a root}\right) = \frac{Z(\beta^{-\mathbf{in}(v)})}{Z(\beta)} \tag{3.97}$$

$$p\left(I_v \text{ is a leaf}\right) = \frac{Z(\beta^{-\mathbf{out}(v)})}{Z(\beta)} \tag{3.98}$$

where $\mathbf{in}(v)$ and $\mathbf{out}(v)$ return the set of all edges in and out of time-series v respectively. $\beta^{-\mathbf{e}}$ indicates all elements of β which involve any edge in the set $\mathbf{e}$ are zero.

The indicator variables used in the examples above can be expressed as a general multiplicative functions of the form

$$g(\bar{E}) = \prod_{v=1}^{N} g_{\mathbf{pa}(v),v}. \tag{3.99}$$

The expected value of a general multiplicative function can be calculated by:

$$\mathbb{E}\left[g(\bar{E})\right] = \sum_{\bar{E}} p_0\left(\bar{E}\right) g(\bar{E}) \tag{3.100}$$

$$= \sum_{\bar{E}} \frac{1}{Z(\beta)} \prod_{i=1}^{N} \beta_{\mathbf{pa}(i),i} \prod_{j=1}^{N} g_{\mathbf{pa}(j),j} \tag{3.101}$$

$$= \frac{1}{Z(\beta)} \sum_{\bar{E}} \prod_{v=1}^{N} \beta_{\mathbf{pa}(v),v} g_{\mathbf{pa}(v),v} \tag{3.102}$$

$$= \frac{Z(\beta \circ g)}{Z(\beta)} \tag{3.103}$$

Note that variance or other higher order moments of multiplicative functions can also calculated in this manner (*e.g.* using $Z(\beta \circ g^2)$ in calculating posterior variance).

In addition, one can calculate the expectation of additive functions of the form

$$f(\bar{E}) = \sum_{v=1}^{N} f_{\mathbf{pa}(v),v}. \tag{3.104}$$

Additive functions allow calculation of quantities such as the expected number of children or parents of a particular time-series. For example, by setting $f_{\mathbf{S},v}$ to $|\mathbf{S}|$ for a single v and all other $f_{.,u\neq v} = 0$, $f(\bar{E})$ will count the number of parents vertices v has in structure $\bar{E}$.

For $\bar{E} \in \mathcal{A}$ or $\bar{E} \in \mathcal{B}_M$ the expectation takes the form:

$$\mathbb{E}\left[f(\bar{E})\right] = \sum_{j=1}^{N} \frac{\gamma_j(\beta \circ f)}{\gamma_j(\beta)} \tag{3.105}$$

For directed trees, $\bar{E} \in \mathcal{T}$,

$$\mathbb{E}\left[f(\bar{E})\right] = \sum_{r=1}^{N} \frac{Z_r(\beta)}{Z(\beta)} \mathrm{tr}\left(M_{r,r}\left(\bar{Q}(\beta \circ f)\right) M_{r,r}\left(\bar{Q}(\beta)\right)^{-1}\right) \tag{3.106}$$

where $M_{i,j}(M)$ is the matrix M with its ith row and jth column removed. A similar form is obtained for directed forests. Derivations of Equations 3.105 and 3.106 can be found in Appendix B.3.

Maximum a Posteriori Structure

If a point estimate of structure is desired, using prior information, one can turn to a MAP estimate. It is easy to see from Equation 3.95 that the MAP directed structure $\bar{E}$ is obtained via

$$\bar{E}^* = \arg\max_{\bar{E}} \prod_{v=1}^{N} \beta_{\mathbf{pa}(v,\bar{E}),v} W_{\mathbf{pa}(v,\bar{E}),v}. \tag{3.107}$$

When $\bar{E} \in \mathcal{A}_N$ or $\bar{E} \in \mathcal{P}_N^K$ each vertex v can be treated independently. That is, $\bar{E}^*$ can be found by N independent optimizations, each of which finds the maximum $\beta_{\mathbf{S},v} W_{\mathbf{S},v}$ over all parent sets $\mathbf{S}$ for time-series v.

For directed forests and trees the optimization becomes more complex due to the global acyclic constraint on $\bar{E}$. It is equivalent to the max weighted directed tree problem. Solutions to this problem were developed independently by Chu and Liu [18], Edmonds [25] and Bock [10]. Appendix A.3.1 gives the details of their algorithm.

Algorithm for Bayesian Inference over Structure

In the previous sections we described how to perform exact Bayesian inference on the structure $\bar{E}$ given observed data $\mathcal{D}$ using a TIM. Algorithm 3.4.1 summarize the necessary steps when analyzing N time-series.

It is important to note that each evidence term $W_{\mathbf{S},v}$ in Equation 3.89 is calculated using the same data $\mathcal{D}$. In Section 3.3.2, we discussed how the ability to exploit parametric differences is lost when point estimates of parameters are obtained from a single $\mathcal{D}$. Additionally, issues arose when comparing increasing expressive structures using these point estimates of parameters. Equation 3.89 avoids some of these issues by integrating over all parameters rather than obtaining point estimates. In particular, the integral has a built in penalty for larger parent sets which would result in more

Algorithm 3.4.1 Bayesian Inference of TIM Structure for Static Dependence Analysis

Require: Observations of N time-series in $\mathcal{D}$.

% Define a TIM

Choose r

Choose the set in which $\bar{E}$ belongs. (*e.g.* $\bar{E} \in \mathcal{A}_N$).

Choose a parameterization for each $p\left(\mathbf{x}_t^v | \widetilde{\mathbf{x}}_t^{v,\mathbf{pa}(v)}, \Theta_{v|\mathbf{pa}(v)}\right)$.

Choose a conjugate prior $p_0\left(\Theta_{v|\mathbf{pa}(v)} | \Upsilon\right)$ for each conditional.

Using these prior terms to form $p_0\left(\Theta|\bar{E}\right)$ using Equation 3.63.

Choose βs and form $p_0\left(\bar{E}\right)$ using Equation 3.64.

% Get the posterior

for all $v \in \{1, \ldots, N\}$ **do**

  for all valid parent sets **S do**

    Calculate $W_{\mathbf{S},v}$ using Equation 3.89

  end for

end for

Use Equation 3.95 to form the posterior $p\left(E|\mathcal{D}\right)$

% Output desired form of result

if A point estimate is desired **then**

  Calculate and report the MAP estimate of $\bar{E}$

else if Posterior structural events, expectations and/or marginal probabilities are desired **then**

  Report them using Equations 3.96 through 3.106

else

  Report the full posterior $p\left(E|\mathcal{D}\right)$

end if

expressive structures. That is, the integral is over the full space of parameters given a parent set **S**. More elements in **S** result in a larger space of parameters to integrate over. The posterior over this space must integrate to 1 and thus each unormalized posterior term in the integral contributes less to the evidence as $|\mathbf{S}|$ increases. However, it remains the case that even this Bayesian approach loses something by using the same data $\mathcal{D}$ to calculate evidence for all substructures. Knowing the true parameters which

generated the data will always yield a better result. That, is if the true underlying parameters, $\Theta_{v|\mathbf{S}}$, were known we could simply use $W_{\mathbf{S},v} = p\left(\mathcal{D}^v|\mathcal{D}^{v,\mathbf{S}}, \Theta_{v|\mathbf{S}}\right)$. It is important to note that in the Bayesian context we do not assume the exists of a "true set" of parameters, by instead a distribution over them with each new draw of $\mathcal{D}$ potentially using a different Θ drawn from this distribution.

■ 3.5 Summary

In this chapter we introduced the general class of static dependence models which are specified by fixed dependence structure and associated parameters. We discussed the main challenges associated with performing structural inference on this class of models, particularly when the parameters are unknown and treated as a nuisance. Namely, the challenge of specifying priors on a large set of structures and integrating over unknown parameters. Two different specific static dependence models were introduced. The challenges associated with inference were addressed in a different way for each model.

A FactM describes time-series in terms of independent groupings. We will use such models for reasoning over a finite set of associations, thus keeping the space of structures tractable. In the absence of prior information we presented an ML approach to structural inference and showed how it is an approximation to the MAP solution which avoids integrating over unknown parameters. An alternative hypothesis testing view of ML inference was presented which exposes the role of structure and parametric differences when choosing among FactMs. This analysis allowed us to characterize what is lost when performing ML inference. That is, it was shown that parametric differences cannot be exploited.

A TIM uses a directed structure to specify more detailed causal relationships among time-series. A conjugate prior on structure and parameters was presented which allowed for exact Bayesian inference using a TIM. This prior allows one to reason over a set of structures which is super-exponential in the number of time-series in exponential-time, in general. Furthermore, it was shown that by imposing simple local or global structural constraints, computation was reduced to polynomial-time complexity. The ability to calculate the exact posterior further allows one to calculate exact marginal posterior event probabilities and statistics.

Note that, while we presented two different approaches for inference for these two models, it is straightforward to define appropriate priors and perform exact Bayesian

inference using a FactM and one can easily perform ML estimation on a TIM. The specific inference technique we chose to present for each model is directly a consequence of the applications in which these models will be used in Chapters 5 and 6.

Chapter 4

Dynamic Dependence Models For Time-Series

In Chapter 3, we presented two static dependence models and examined how they could be used to analyze the dependence relationships among multiple time-series. These models assumed that the dependence relationship among time-series was fixed over all time. In this chapter, we remove the static structure assumption and look at the problem of *dynamic dependence analysis*. That is, given multiple time-series we wish to analyze the way in which their dependence structure evolves over time. As we will see, incorporating the notion of dynamically evolving dependence structures complicates inference. In the dynamic framework, we consider both point estimates and a full characterizations of posterior uncertainty on structure over time.

As discussed in Chapter 3, the number of possible structures used in a static dependence model generally grows super-exponentially with the number of time-series being analyzed. An obvious challenge for a dynamic dependence analysis tool is that, by allowing the structure to change over time, the number of possible *sequences* of structure increase exponentially as the number of time points grows. That is, if on considers S allowable structures at each time point, there are S^T possible sequences of structure over a period of T time points.

In this chapter, we discuss ways to efficiently reason over these sequences of changing dependence relationships. We assume dependence relationships have some temporal persistence and that they may be revisited over time. These two assumptions are useful in that they simplify aspects of inference and provide benefits in terms of estimation quality. Two applications are explored in Chapters 5 and 6 in which both assumptions are reasonable. For example, in an audio-visual speech association task it is common to assume that individuals produce continuous speech segments rather than short bursts

of sound. It is also expected that each person will speak multiple times during a conversation.

We begin in Section 4.1 with a discussion of standard windowed approaches for dynamic dependence analysis. Next, we introduce the concept of a *dynamic dependence model* in Section 4.2. These models extend the static dependence models presented in Chapter 3 via the introduction of a dynamically evolving latent state variable. The state variable indexes structure, allowing switching among a finite set of dependence relationships over time. In Section 4.3, we detail inference using a dynamic dependence model in a maximum likelihood setting in which a point estimate of the sequence of active structures is desired. Following a similar analysis to that presented in Section 3.3.2 we show that, in contrast to static dependence models, dynamic dependence models allow one to take advantage of parametric differences to help identify structure. A set of illustrative examples is provided in Section 4.2.1. In Section 4.4 we introduce a tractable prior for dynamic dependence models and discuss Bayesian inference of structure sequences. Lastly, in Section 4.5 we discuss our model in the context of related work.

■ 4.1 Windowed Approaches

Using our assumption that dependence relationships do not switch rapidly, one approach for performing dynamic dependence analysis is to treat the problem as a series of static dependence inference tasks. That is, one can start by assuming the structure is fixed within a window of time $\mathbf{t}_w = [t - w/2, \ldots, t, \ldots t + w/2]$. Within this window, static dependence analysis as described in Chapter 3 can be performed on $\mathcal{D}_{\mathbf{t}_w}$. In order to deal with the structure changing over time, this window is moved forward to a future time point $t+\delta$ and analysis is repeated. Such an approach treats each window independently, transforming the inference problem from one of inference over an exponential number of *sequences* of structure to one which is linear in the number windows.

There are two main issues that one must consider when using a windowed approach. First is the open question of what window size to pick. There is a bias-variance tradeoff as a function of window size. That is, long windows bias your result since they more likely to contain times in which structure changes. Within shorter windows structure is more likely to be stationary, but as a consequence of using fewer samples the estimate of this structure is less accurate / has higher variance. In fact, there is no fixed window

size that is guaranteed to never violate the static dependence assumption. A fixed length window will eventually overlap a change of structure as it is moved forward over time.

A second issue with windowed approaches is that they inherit the challenges associated with structural inference for static dependence models due to their myopic windowed view of data. That is, the inference performed for a particular window only uses information from the data observed within that window. By definition, a windowed approach cannot take advantage of information from data observed in the past or future which may share the same structure and/or parameters as the data observed within the window analyzed. As discussed in Section 3.3.2, parametric differences can improve structural inference due to increased separability of hypothesized models. In a windowed approach, parameters for different structures are estimated or integrated over using the same data for each hypothesized structure. Thus, such an approach cannot uncover or exploit the true underlying parametric differences. In addition, if a maximum likelihood approach is utilized, significance must also be estimated when reasoning over nested hypothesized structures. That is, one can always increase the likelihood by using a more complex model and thus one must turn to estimation of significance when deciding if a less complex model should be rejected. These issues are explored and discussed using simple illustrative experiments in Section 4.3.2

■ 4.2 Dynamic Dependence Models

In this section, we present the concept of a *dynamic dependence model* (DDM). A DDM explicitly models evolving dependence structure among time-series. We will show that dynamic dependence analysis can be mapped to inference using this model rather than to a series of independent static dependence inference tasks. A DDM can be represented as a dynamic Bayesian network as depicted in Figure 4.1. A hidden discrete state at time t, z_t, indexes one of K specific structures, E^{z_t}, and parameters, Θ^{z_t}. The observed time-series at time t are modeled using a static dependence model, as presented in Chapter 3, with structure specified by E^{z_t} and parameters specified by Θ^{z_t}. Assuming r-th order temporal dependence and K possible states, the generative model for a

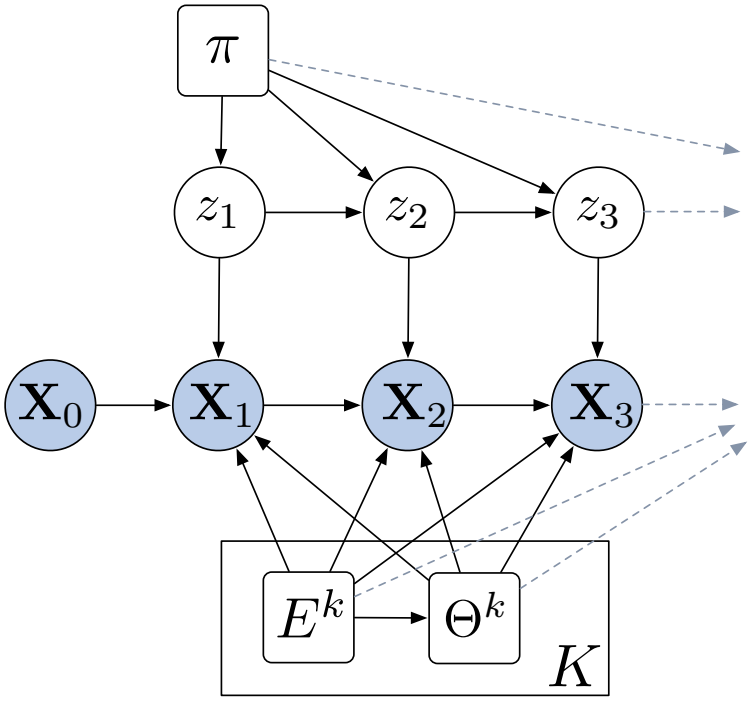


Figure 4.1. *First Order Dynamic Dependence Model:* A graphical model representing a generic first order ($r = 1$) dynamic dependence model is shown. The square box around the structure E^k and parameters Θ^k is a plate indicating there are K independent copies of these parameters and structure. π is the set of parameters describing the transition probabilities for evolving latent state z.

DDM(r,K) over a time period $\mathbf{t} = \{1, \ldots, T\}$ takes has the following form:

$$p\left(\mathbf{X_t}, z_\mathbf{t} | \mathbf{E}, \mathbf{\Theta}\right) = p\left(\mathbf{X_t} | z_\mathbf{t}, \mathbf{E}, \mathbf{\Theta}\right) p\left(z_\mathbf{t}\right) \tag{4.1}$$

$$= \prod_{t=1}^{T} p\left(\mathbf{X}_t | \widetilde{\mathbf{X}}_t, E^{z_t}, \Theta^{z_t}\right) p\left(z_t | \pi^{z_{t-1}}\right), \tag{4.2}$$

where $z_0 \triangleq 0$, $\mathbf{E} = \{E^1, \ldots, E^K\}$ is a set of structures, and $\mathbf{\Theta} = \{\pi^0, \ldots, \pi^K, \Theta^1, \ldots, \Theta^K\}$ is a set of all parameters for the model. The state sequence, $z_{1:T}$, is modeled as a first order Markov process with a discrete transition distribution,

$$p\left(z_t | z_{t-1}, \mathbf{\Theta}\right) = p\left(z_t | \pi^{z_{t-1}}\right) \tag{4.3}$$

$$= \pi_{z_t}^{z_{t-1}}. \tag{4.4}$$

The parameters $\pi^j = \{\pi_1^j, \ldots, \pi_K^j\}$ are the transition probabilities to all K states from state j. Each state k indexes an r-th order static dependence model which defines

the form of $p\left(\mathbf{X}_t|\widetilde{\mathbf{X}}_t, E^k, \Theta^k\right)$. If a FactM is used with the structure specified in terms of a factorization/hyperedges F, we refer to the DDM as a *hidden factorization Markov model*, HFactMM(r,K) [81]. If a TIM is used with structure specified in terms of directed edges $\bar{E}$, we refer to the DDM as a *switching temporal interaction model*, STIM(r,K) [82].

Note that a DDM assumes that the number of states, K, is known and that structures (and parameters) can be revisited over time. We note that the static dependence models presented in Chapter 3 can also easily be embedded into other alternative dynamic models. For example, one may use a parts partition model (PPM) similar to that used by Xuan and Murphy [94]. A PPM allows for an unknown number of states that are never revisited. Similarly, one may adopt nonparametric Bayesian models such as the hierarchical Dirichlet process hidden Markov model (HDP-HMM)[87] to allow for an unknown number of potential revisited states. The appropriate choice of model is highly dependent on the how well the underlying assumptions of the model match the application of interest. While there are interesting details within each of these modeling choices, here, we focus on dynamic estimation when the number states is know and each state is likely to be revisited. That is, the applications presented in this dissertation focus on the use of DDMs, leaving the above straightforward modifications/extensions for future work.

■ 4.2.1 Dynamic Dependence Analysis Using a DDM

We now turn to the core task of dynamic dependence analysis. Given observations $\mathcal{D}$ of multiple time-series for over a period of time $\mathbf{t} = \{1, \ldots, T\}$, the goal of dynamic dependence analysis is to characterize the dependence among these time-series at each point in time $t \in \mathbf{t}$. Dynamic dependence analysis can be carried out using a DDM. That is, using a DDM we are interested in inferring information about $z_{1:T}$ and $\mathbf{E}$ given observed data. The set $\mathbf{E}$ contains the dependence structure for each of the K states and $z_{1:T}$ indicates which state is active at each point in time. If prior information is available, ideally, one would like to calculate the posterior

$$p\left(z_{1:T}, \mathbf{E}|\mathcal{D}\right) = \int p_0\left(z_{1:T}, \mathbf{E}, \mathbf{\Theta}|\mathcal{D}\right) d\mathbf{\Theta}. \tag{4.5}$$

From this joint posterior on the set of structures $\mathbf{E}$ and state sequence $z_{1:T}$, a wide variety of useful statistics can be calculated. For example, the MAP state sequence and structures can be obtained. Unfortunately, no simple analytic form exists for Equation

4.5 and brute force calculation is intractable due to the size of the joint space. There are K^T possible state sequences, $z_{1:T}$, and potentially a super-exponential number of structures for each of the K elements, E^k, in $\mathbf{E}$.

Fortunately, however, DDMs have a well studied structure. They are nothing more than specialized hidden Markov models (HMMs) or switching vector autoregressive (SVAR) models. They differ from these standard models in that the form of the dependence among the observations is controlled by the *values* of the hidden structure $\mathbf{E}$[1]. In the following two sections we discuss both maximum likelihood and Bayesian approaches to structural inference using a DDM. Both approaches will take advantage of two key properties of the model:

1. The likelihood $p(\mathcal{D}|\mathbf{E}, \mathbf{\Theta})$ as well as the posterior on each state z_t given the data and all other unobserved random variables, $p(z_t|\mathcal{D}, \mathbf{E}, \mathbf{\Theta})$, can be calculated tractably using standard forward-backward message passing [4].

2. Given the state sequence $z_{1:T}$ one can pool information from time points that share a common state k in order to help one infer the structure E^k and parameters Θ^k.

■ 4.3 Maximum Likelihood Inference

In the absence of prior information, one can adopt a classical maximum likelihood approach for estimating the unobserved structures $\mathbf{E}$ and parameters $\mathbf{\Theta}$. That is, similar to the approach presented in Section 3.3.2, we wish to find

$$\{\hat{\mathbf{E}}, \hat{\mathbf{\Theta}}\} = \arg\max_{\mathbf{E},\mathbf{\Theta}} p(\mathcal{D}|\mathbf{E}, \mathbf{\Theta}) \tag{4.6}$$

$$= \arg\max_{\mathbf{E},\mathbf{\Theta}} \sum_{z_{1:T}} p(\mathcal{D}|z_{1:T}, \mathbf{E}, \mathbf{\Theta})\, p_0(z_{1:T}|\mathbf{\Theta}). \tag{4.7}$$

Given an estimate of the structures $\hat{\mathbf{E}}$ and parameters $\hat{\mathbf{\Theta}}$, one can then find the MAP state sequence:

$$\hat{z}_{1:T} = \arg\max_{z_{1:T}} p\left(z_{1:T}|\mathcal{D}, \hat{\mathbf{E}}, \hat{\mathbf{\Theta}}\right). \tag{4.8}$$

Again, as discussed in Section 3.3.2, such an approach is related to generalized likelihood methods for dealing with nuisance parameters. We will discuss the details of the optimization in Equation 4.7 in following section along with an analysis of how state sequences distinguish themselves from each other when calculating Equation 4.8.

[1] We will put our model in the context of previous work in Section 4.5

4.3.1 Expectation Maximization

The optimization in Equation 4.7 is complicated by the fact that one must sum over all K^T possible state sequences and no closed form analytical solution exists. In this dissertation, we use the standard approach of Expectation Maximization (EM) to find a local maxima of $p(\mathcal{D}|\mathbf{E}, \mathbf{\Theta})$ [23]. Given an initial estimate of the structure and parameters, $\mathbf{E}^{(0)}$ and $\mathbf{\Theta}^{(0)}$, the EM algorithm iteratively updates its estimate by repeating two basic steps. For iteration i, these two steps are:

1. E-Step: Find a function which is a tight lower bound for $p\left(\mathcal{D}|\mathbf{E}^{(i-1)}, \mathbf{\Theta}^{(i-1)}\right)$.
2. M-Step: Given this function, find a new set of structures and parameters, $\mathbf{E}^{(i)}, \mathbf{\Theta}^{(i)}$, which maximize it.

(c.f. [65]). For a DDM (or general HMM or SVAR), the function which provides the tight bound is simply the posterior $p\left(z_{1:T}|\mathcal{D}, \mathbf{E}^{(i-1)}, \mathbf{\Theta}^{(i-1)}\right)$. The M-Step is a series of K independent maximization problems each of the form:

$$\{E^{k(i)}, \Theta^{k(i)}\} = \arg\max_{E^k, \Theta^k} \sum_{t=1}^{T} \gamma_t^k \log p\left(\mathcal{D}_t|\widetilde{\mathcal{D}}_t, E^k, \Theta^k\right) \tag{4.9}$$

where γ_t^k was calculated in the E-Step using standard forward-backward message passing [4, 76] such that

$$\gamma_t^k = p\left(z_t = k|\mathcal{D}, \mathbf{E}^{(i-1)}, \mathbf{\Theta}^{(i-1)}\right). \tag{4.10}$$

That is, the M-Step does a weighted ML estimate of structure and parameters for each state. The weights are a function of the posterior probability of being in state k at each point in time. For a DDM, the weighted ML step is performed on a static dependence model. For example, if one is using an HFactMM, the optimization in Equation 4.9 would be a simple modification of Equation 3.19 in Section 3.3.2. That is, for each state k, the M-Step is:

$$\left\{\hat{F}^k, \hat{\Theta}^k\right\} = \arg\max_{F, \Theta} \sum_{t=1}^{T} \gamma_t^k \sum_{f=1}^{|F|} \log p\left(\mathcal{D}_t^{F_f}|\widetilde{\mathcal{D}}_t^{F_f}, \Theta_{F_f}\right) \tag{4.11}$$

$$= \arg\max_{F} \sum_{f=1}^{|F|} \arg\max_{\Theta_{F_f}} \sum_{t=1}^{T} \gamma_t^k \log p\left(\mathcal{D}_t^{F_f}|\widetilde{\mathcal{D}}_t^{F_f}, \Theta_{F_f}\right) \tag{4.12}$$

where we have switched to the notation F to indicate we are specifically discussing ML inference on an HFactMM here. This optimization can be expressed as two steps. First, one must find the maximum weighted likelihood for each factor:

$$\hat{\Theta}_{F_f} = \arg\max_{\Theta_{F_f}} \sum_{t=1}^{T} \gamma_t^k \log p\left(\mathcal{D}_t^{F_f} | \widetilde{\mathcal{D}}_t^{F_f}, \Theta_{F_f}\right) \tag{4.13}$$

$$= \arg\max_{\Theta_{F_f}} W_{F_f}^{\gamma}(\Theta_{F_f}) \tag{4.14}$$

For Gaussian distributions the parameters $\hat{\Theta}_{F_f}$ will be weighted estimates of the mean and covariance, while for discrete distributions weighted counts will be used for estimating symbol probabilities. Given $W_{F_f}^{\gamma}(\hat{\Theta}_{F_f})$ for every allowable factor F_f, Equation 4.12 becomes

$$\left\{\hat{F}^k, \hat{\Theta}^k\right\} = \arg\max_{F,\Theta} \sum_{f=1}^{|F|} W_{F_f}^{\gamma}(\Theta_{F_f}). \tag{4.15}$$

In addition to estimating structure and parameters Θ, the M-Step also updates the transition distribution parameters $\pi^0, \ldots \pi^k$. The E-step and M-step are iterated until convergence. Convergence can be defined in terms of stable parameter estimates or overall likelihood of the data given the current parameter estimates. A summary of the EM algorithm for a DDM is given in Algorithm 4.3.1 and forward backward message passing is outlined in Algorithm 4.3.2 . In practice we run multiple EM optimizations, each starting with a different random initialization. The parameters with the highest likelihood over all initializations is taken as the final result. We refer the reader to [76] for a basic tutorial on EM for HMMs.

Analysis of Parametric Differences

The MAP state sequence in Equation 4.8 can be calculated efficiently via dynamic programing using the Viterbi algorithm [92, 31]. Viterbi decoding implicitly performs an M-ary hypothesis test comparing all $M = K^T$ possible state sequence. Much like the analysis shown in Section 3.3.2, this alternative hypothesis testing view helps expose how state sequences distinguish themselves from each other.

We examine how state sequences drawn from an HFactMM(0,K) with corresponding estimates of structures $\hat{\mathbf{F}}$ and parameters $\hat{\mathbf{\Theta}}$ distinguish themselves from each other. We examine the case in which $r = 0$ here for simplicity and to match with our experiments

Algorithm 4.3.1 The EM Algorithm for a Dynamic Dependence Model

Require: N observed time-series from $1:T$ in $\mathcal{D}$, and specific parameterization/choice of a DDM.

% Initialize

Randomly set $\mathbf{E}^{(0)}$ and $\mathbf{\Theta}^{(0)}$

$i \leftarrow 0$

% Expectation Maximization

repeat

$i \leftarrow i + 1$

% E-Step

$\{\gamma, \xi\} \leftarrow$ ForwardBackward$(\mathcal{D}, \mathbf{E}^{(i-1)}, \mathbf{\Theta}^{(i-1)}, K)$ **% See Algorithm 4.3.2**

% M-Step

$\left\{\pi_1^0, \ldots, \pi_K^0\right\}^{(i)} \leftarrow \left\{\gamma_1^1, \ldots, \gamma_1^K\right\}$

for $k = 1$ to K **do**

$\left\{E^{k(i)}, \Theta^{k(i)}\right\} \leftarrow \arg\max_{E,\Theta} \sum_{t=1}^{T} \gamma_t^k \log p\left(\mathcal{D}_t | \widetilde{\mathcal{D}}_t, E, \Theta\right)$

for $j = 1$ to K **do**

$\pi_j^{k(i)} \leftarrow \sum_{t=1}^{T-1} \xi_t^{k,j} / \sum_{t=1}^{T-1} \gamma_t^k$

end for

end for

until Convergence

% Report Results

Output $\mathbf{E}^{(i)}$ and $\mathbf{\Theta}^{(i)}$ along with a calculation of $p\left(\mathcal{D} | \mathbf{E}^{(i)}, \mathbf{\Theta}^{(i)}\right)$.

in Chapter 5. Consider a binary hypothesis test between two different state sequences in which

$$H_1 : z_{1:T} = a_{1:T} \tag{4.16}$$

$$H_2 : z_{1:T} = b_{1:T} \tag{4.17}$$

$$\tag{4.18}$$

We define a common factorization, $F^{\cap_t}$, in similar manner to that shown in Section 3.3.2. That is , $F^{\cap_t}$ is the factorization common to the factorizations F^{a_t} and F^{b_t}.

Algorithm 4.3.2 The Forward-Backward Algorithm

function FORWARDBACKWARD($\mathcal{D}, \mathbf{E}, \mathbf{\Theta}, K$)

Let $p_t^k \triangleq p\left(\mathcal{D}_t|\widetilde{\mathcal{D}}_t, E^k, \Theta^k\right)$

% **Forward:** $\alpha_t^k = p\left(\mathcal{D}_{1:t}, z_t = k|\mathbf{E}, \mathbf{\Theta}\right)$

for $k = 1$ to K **do**

$\alpha_1^k \leftarrow p_t^k \pi_k^0$ % **Initialize forward messages**

end for

for $t = 1$ to $T - 1$ **do**

for $k = 1$ to K **do**

$\alpha_{t+1}^k \leftarrow \left(\sum_{j=1}^K \alpha_t^j \pi_k^j\right) p_{t+1}^k$

end for

end for

% **Backward:** $\beta_t^k = p\left(\mathcal{D}_{(t+1):T}|z_t = k, \mathbf{E}, \mathbf{\Theta}\right)$

for $k = 1$ to K **do**

$\beta_T^k \leftarrow 1$ % **Initialize backward messages**

end for

for $t = T - 1$ down to 1 **do**

for $k = 1$ to K **do**

$\beta_t^k \leftarrow \sum_{j=1}^K \pi_j^k p_{t+1}^j \beta_{t+1}^j$

end for

end for

% **Calc** $\gamma_t^k = p\left(z_t = k|\mathcal{D}, \mathbf{E}, \mathbf{\Theta}\right)$ **and** $\xi_t^{k,j} = p\left(z_t = k, z_{t+1} = j|\mathcal{D}, \mathbf{E}, \mathbf{\Theta}\right)$

for $t = 1$ to $T - 1$ **do**

for $k = 1$ to K **do**

$\gamma_t^k \leftarrow \alpha_t^k \beta_t^k / \left(\sum_{j=1}^K \alpha_t^j \beta_t^j\right)$

for $j = 1$ to K **do**

$\xi_t^{k,j} \leftarrow \left(\alpha_t^k \pi_j^k p_{t+1}^j \beta_{t+1}^j\right) / \left(\sum_{m=1}^K \sum_{n=1}^k \alpha_t^m \pi_n^m p_{t+1}^n\right)$

end for

end for

end for

return γ and ξ

end function

Given learned parameters and structures the hypothesis test takes following form:

$$\hat{l}_{1,2} \triangleq \log \frac{p\left(\mathcal{D}|a_{1:T}, \hat{\mathbf{F}}, \hat{\boldsymbol{\Theta}}\right)}{p\left(\mathcal{D}|b_{1:T}, \hat{\mathbf{F}}, \hat{\boldsymbol{\Theta}}\right)} \underset{H2}{\overset{H1}{\gtrless}} \log \frac{p\left(a_{1:T}|\hat{\boldsymbol{\Theta}}\right)}{p\left(b_{1:T}|\hat{\boldsymbol{\Theta}}\right)}. \tag{4.19}$$

The test compares the log likelihood ratio to a threshold formed from comparing the dynamics of the two hypothesized state sequences. Using the same form as Equation 3.36 one can show that in expectation under H_1, with $r = 0$, the likelihood ratio is

$$\begin{aligned}\mathbb{E}_{\mathcal{D}}\left[\hat{l}_{1,2}|H_1\right] = &\sum_{t\in\mathbf{d}} D\left(\; p\left(\mathcal{D}_t|\hat{F}^{a_t}, \hat{\Theta}^{a_t}\right) \middle\| p\left(\mathcal{D}_t|F^{\cap_t}, \hat{\Theta}^{a_t}\right) \;\right) \\ &+ \sum_{t\in\mathbf{d}} D\left(\; p\left(\mathcal{D}_t|F^{\cap_t}, \hat{\Theta}^{a_t}\right) \middle\| p\left(\mathcal{D}_t|\hat{F}^{b_t}, \hat{\Theta}^{b_t}\right) \;\right)\end{aligned} \tag{4.20}$$

and similarly when H_2 is true

$$\begin{aligned}\mathbb{E}_{\mathcal{D}}\left[\hat{l}_{1,2}|H_2\right] = &-\sum_{t\in\mathbf{d}} D\left(\; p\left(\mathcal{D}_t|\hat{F}^{b_t}, \hat{\Theta}^{b_t}\right) \middle\| p\left(\mathcal{D}_t|F^{\cap_t}, \hat{\Theta}^{b_t}\right) \;\right) \\ &- \sum_{t\in\mathbf{d}} D\left(\; p\left(\mathcal{D}_t|F^{\cap_t}, \hat{\Theta}^{b_t}\right) \middle\| p\left(\mathcal{D}_t|\hat{F}^{a_t}, \hat{\Theta}^{a_t}\right) \;\right)\end{aligned} \tag{4.21}$$

where $\mathbf{d} = \{t \mid a_t \neq b_t\}$ is the set of all time points in which the hypothesized state sequences disagree. See Appendix B.4 for details. Much like in the static case, we see that the expected log likelihood ratio can be decomposed into two terms. The first term compares structure of the true hypothesis to the common structure under a consistent set of parameters. The second term exploits both structural and parametric differences.

However, unlike the situation in the static case, the parameters $\hat{\Theta}^k$ are estimated via the weighted maximum likelihood estimate in Equation 4.9. Each parameter is estimated using the observed data differently and thus parametric differences can be exploited. That is, the second set of terms will not drop out as they did in the static case as discussed in Section 3.3.2.

■ 4.3.2 Illustrative Examples

In the previous section we discussed how ML inference using an HFactMM can exploit both structural and parameter differences among the time-series to help identify changes in their dependence relationships. In this section, we present a set of simple illustrative experiments in order to provide some more intuition about these models and their uses. That is, we present experiments to help answer the following questions:

1. How much can be gained by using an HFactMM over standard window approaches? That is, how is the performance of both windowed analysis and ML inference using a HFactMM affected as we change both the structural and parametric differences between the dependence relationships found in the data.

2. When are state dynamics important?

3. What can be done when the correct model parameterization is unknown?

We begin with a simple experiment involving two (*i.e.* $N = 2$) 1-d time-series. We create a HFactMM($r = 0$,$K = 2$) with the factorizations for each state set to be independent $F^1 = \{\{1\},\{2\}\}$ and dependent $F^2 = \{\{1,2\}\}$. Since $r = 0$, this model produces i.i.d. samples. Gaussian factor models are chosen such that parameters for state $k = 1$, Θ^1_1, Θ^1_2, describe zero mean, unit variance distributions. The parameters for state 2, $\Theta^2_{1,2}$, are set such that the joint mean is $[0\ \Delta]^\mathsf{T}$, and the covariance is full, with unit marginal variance and correlation ρ. The transition dynamic parameters are set such that starting in either state is equally likely, $\pi^0_1 = \pi^0_2 = .5$, and that there is 0.95 probability of self transition, $\pi^1_1 = \pi^2_2 = 0.95, \pi^1_2 = \pi^2_1 = 0.05$. This yields a model with a simple state dynamic and a control on structural and parametric differences via ρ and D respectively.

For each setting of ρ and Δ, 200 samples (*i.e.* $T = 200$) of the joint process $z_{1:T}$ and $\mathbf{X}_{1:T}$ are drawn. That is, $z_{1:T}$ is drawn using the transitions distribution and then $\mathbf{X}_{1:T}$ is drawn to form observations $\mathcal{D}$ given this state sequence. Figure 4.2 shows one realization. The top of the figure shows $\mathcal{D}$ colored by which state each sample came from along with an indication of how the Gaussian conditional models are parameterized. Δ controls the separation between each conditional FactM distribution and ρ controls the correlation in the FactM used for state 2. The bottom of the figure shows the sampled state sequence as sequence of colors representing which state is active at each point in time.

We compare 3 different approaches for dynamic dependence analysis on this data. The first is an ML windowed approach reasoning over two possible factorization models. The second approach performs ML inference using a HFactM(0,2) with the correct structures given but with unknown parameters. The third performs ML inference using a modified model which has no temporal dynamic on the state sequence. We will refer to this model as a factorization mixture model, FactMM(0,2). Each approach outputs a labeling for each time point indicating whether or not the observations are dependent.

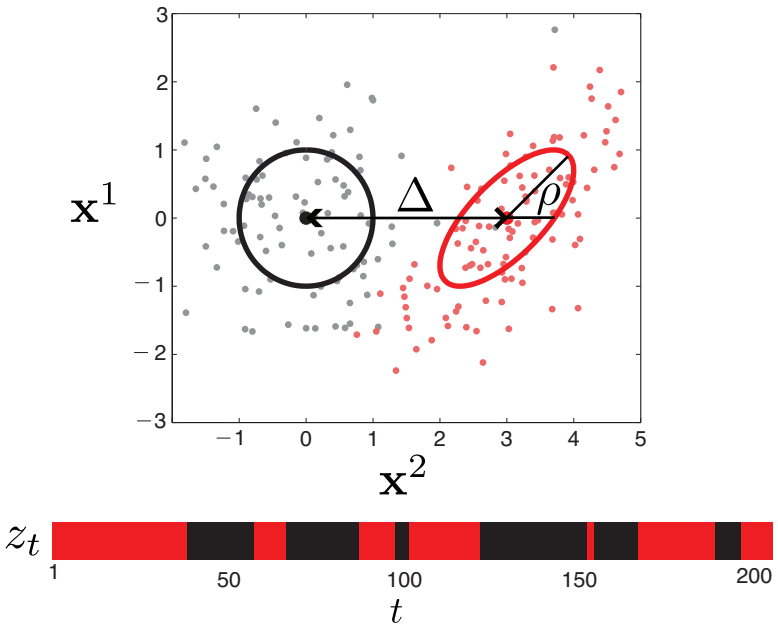


Figure 4.2. *A Sample Draw from an HFactMM(0,2):* The top part of the figure shows the samples colored by their state (black=state 1, red=state 2). The parametric differences between the two possible dependence models is controlled via Δ, and the structural difference controlled by ρ. The bottom part of the figure depicts z_t using the same colors.

This is compared to the known $z_{1:T}$ to calculate performance.

The ML windowed approach reduces to calculating the correlation between observed time-series as shown in the example in Section 3.3.2 and as was the case in that example, the factorizations of interest here are nested. Thus we also estimate significance via calculation of a p-value for each window. Window sizes of 5, 10, 20, and 40 samples were each tested. We record results obtained in an unrealistic, best-case scenario for the windowed analysis. That is, for each data set analyzed, we find the threshold on the p-value that yielded the best performance for each window size and then reported the best result over all window sizes.

Results are shown in Figures 4.3 and 4.4. Each plot shows the average probability of error over 100 trials for various settings of Δ and ρ. Consistent with the analysis presented in Section 3.3.2, Figure 4.3 shows that the performance of the windowed analysis is not affected by changes in the non-structural parameter, Δ. The slight

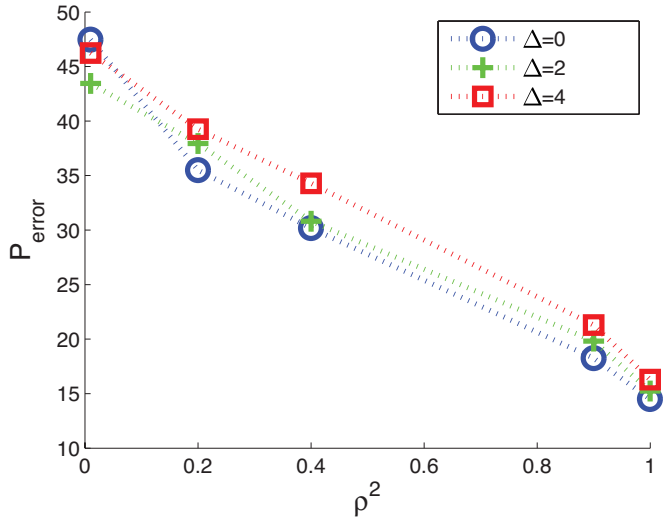


Figure 4.3. *2D Gaussian Experimental Results Using a Windowed Approach:* Performance is measured in terms of average % error over 100 trials. Results are shown as a function ρ for various settings of Δ.

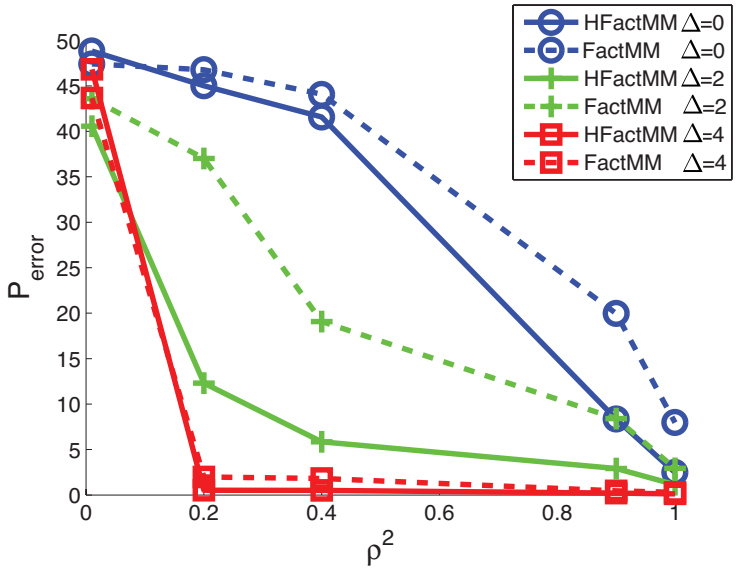


Figure 4.4. *2D Gaussian Experimental results Using an HFactMM and FactMM:* The HFactMM results are shown as solid lines while the FactMM results are shown as dashed lines. Performance is measured in terms of average % error over 100 trials. Results are shown as a function ρ for various settings of Δ.

decreases in performance as Δ increases is due to the fact that larger Δ makes the data look more dependent when a window overlaps a transition between dependent and independent samples.

As predicted by our analysis in Section 4.3 and in contrast to the windowed test, both the HFactMM and FactMM dramatically improve as Δ increases. In general, all approaches improve in performance with increasing ρ, with more rapid improvements for the HFactMM and FactMM for larger Δ. Dynamics help most when Δ is small, *i.e.* when the state conditional distributions overlap. For example, when $\Delta = 2$ the gap in performance between the HFactMM and FactMM is particularly large and there is great benefit to incorporating dynamics by using an HFactMM.

In the previous experiment we used a simple gaussian parameterization for our state conditional distributions/FactMs. More complex parameterizations can be used. However, as the next example shows, certain ambiguities may arise. Such ambiguities can be overcome when an underlying dynamic is present. Consider the data shown in Figure 4.5(a). This data was generated with an HFactMM(0,2) using the same dependent and independent factorizations as in the previous example. However, in this case the factor models are no longer single Gaussians. When the data is independent it is sampled from the product of two Gaussian mixture models, each of which has two mixture components each with spherical unit covariance. The result is a four component mixture model in the joint space (indicated by black circles in the figure). When the data is dependent it comes from a four component joint mixture (indicated by the red circles).

Figures 4.5(b) and 4.5(c) show FactMM and HFactMM models learned from 200 samples of this mixture model using the true parameterization (*i.e.* correct number of mixtures) but unknown parameters. Using mixture models for each state conditional FactM complicates the M-Step in EM slightly. The M-step itself must use an embedded EM step to learn the mixture model for each state.

Note that for this particular model there are many possible combinations of independent and dependent mixtures. In fact, the FactMM model estimated/learned one of these alternative mixtures in Figure 4.5(b). This is because by assuming independent samples and ignoring the state dynamic all valid combinations of dependent and independent mixtures are equally likely. By incorporating dynamics, the HFactMM finds the correct solution. Over 100 trials the HFactMM and FactMM models had an average performance of 64% and 99% accuracy in labeling samples as dependent or not

respectively.

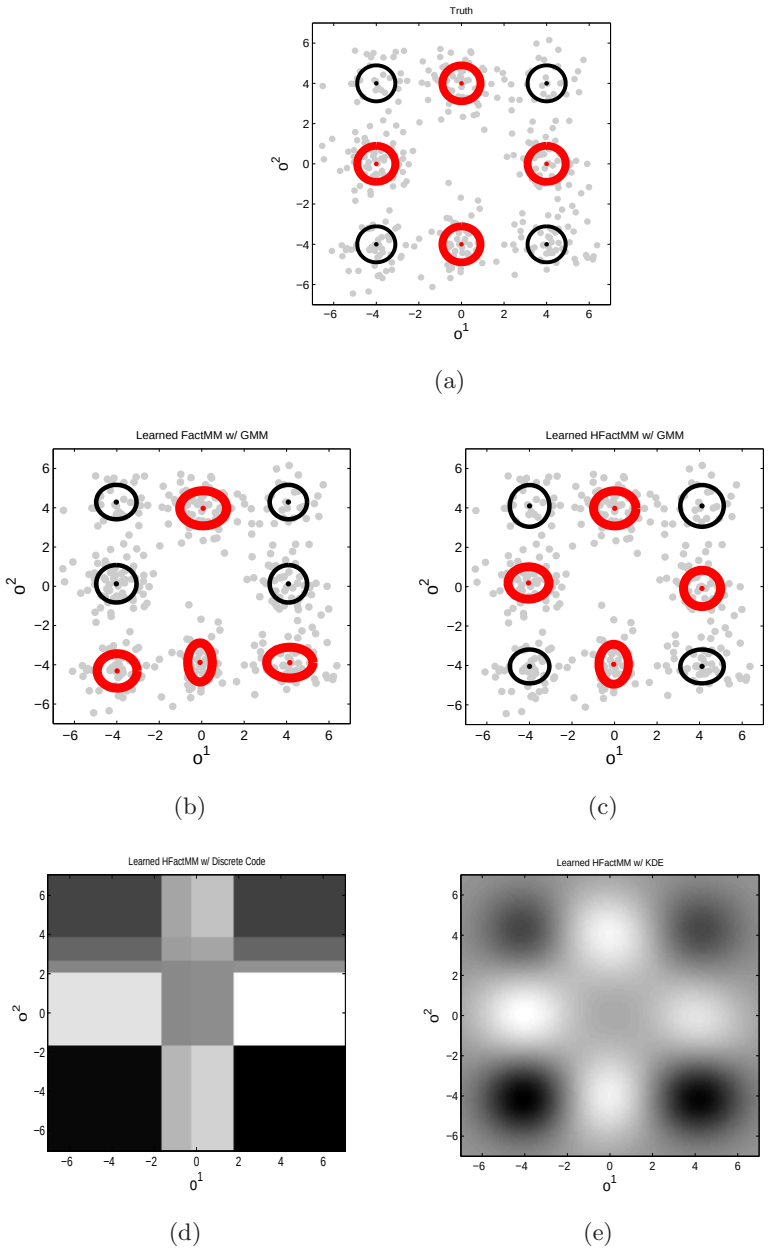
A lingering question is what to do when the "correct" parameterization is not known. One approach is to utilize a nonparametric model for each factor in the state conditional FactMs. A non-parametric sample-based kernel density estimate (KDE) can be used for each factor f of factorization F^k such that:

$$p\left(\mathbf{x}_t^{F_f^k}|\Theta^k\right) = \frac{1}{T}\sum_{j=1}^{T}\gamma_j^k K\left(\mathbf{x}_t^{F_f^k} - \theta_j^{F_f^k}; \sigma^k\right) \tag{4.22}$$

$$= \frac{1}{T}\sum_{j=1}^{T}\gamma_j^k K\left(\mathbf{x}_t^{F_f^k} - \mathbf{x}_j^{F_f^k}; \sigma^k\right) \tag{4.23}$$

where $K()$ is a valid kernel function with a kernel size σ^k and γ_j^k is defined in Equation 4.10. Note here, in addition to specifying σ^k, the parameters Θ are the observations in $\mathcal{D}$, with $\theta_j = \mathbf{X}_j$. Given mixture data used in the previous example, Figure 4.5(e) shows the learned HFactMM model using a KDE with a Gaussian kernel for its state conditional distributions . The figure shows the difference in state conditional distributions at each point in the observation space: $p\left(\mathbf{X}_t|z_t = 2, F^1, \hat{\Theta}^2\right) - p\left(\mathbf{X}_t|z_t = 1, F^1, \hat{\Theta}^1\right)$. Leave-one-out likelihood was used to learn the kernel size σ^k. It is important to note that one must be careful when using more powerful state-conditional distributions. If a single state conditional distribution is flexible enough to describe all of the data and the state transition probabilities are learned an HFactMM can describe the data over all time well using a single state. One way to deal with this issue to modify learning to favor non-degenerate state transition distribution parameters π_k or to limit the complexity of the state conditional models.

An alternative approach to using nonparametric density estimators is to operate on vector quantized versions of each time-series. A separate codebook, C^v, for each observed time-series $\mathcal{D}^v$ is obtained via vector quantization. This can be done using the k-means algorithm or fitting a Gaussian mixture model (cf. [24]) which treats each time point independently. These codebooks are then used to encode the data as discrete time-series $\hat{\mathcal{D}}$. Each time-series is quantized separately prior to dependence analysis. Figure 4.6 depicts this process as signal processing block diagram. The quantized data is modeled with dynamic dependence model with discrete state conditional distributions. Creating an separate eight symbol codebook for each time-series, and then using the corresponding discrete quantized versions each time-series for dynamic dependence analysis using an HFactMM, we estimate/learn parameters which yield the state condi-



(a)

(b)

(c)

(d)

(e)

Figure 4.5. *A More Complex 2D Example:* a) True distribution, F^0=thin black, F^1=thick red b) Learned FactMM w/ correct parameterization c) learned HFactMM with correct parameterization d) learned HFactMM with Discrete Code. e) Learned HFactMM w/ KDE. The mean accuracy over 50 trials was FactMM=64%, HFactMM=99%, HFactMM w/ Discrete Code=98%, HFactMM w/ KDE=98%

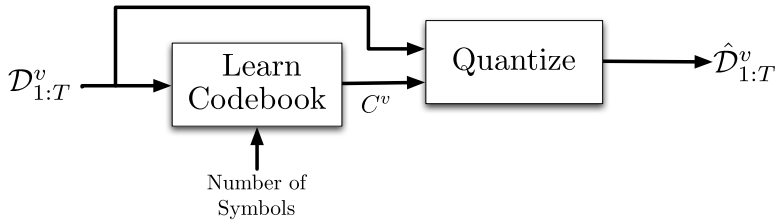


Figure 4.6. *Quantization of Each Observed Time-Series $\mathcal{D}^v$:* A codebook is learned for the input data treating each sample t independently. Using the codebook $\mathcal{D}^v$ is quantized to form a new $\hat{\mathcal{D}}^v$. Note that each time-series is quantized separately.

tional models shown in Figure 4.5(d). This technique produced an average accuracy of 98% in 100 trials. Note that while the model is a crude approximation to the true distribution it captures enough information about dependence to make a correct decision. We will take a similar approach in Chapter 5.

■ 4.4 Bayesian Inference

In the previous section we examined how ML inference can be used to obtain point estimates of K structures and a state sequence indexing which structure is active at each point in time. In this section we return to the problem of obtaining a full posterior on structure and state sequence given the observed time-series rather than point estimates. Again, this is a difficult task due to the number possible state sequences, K^T, and requires one to define priors over the unobserved parameters and structure.

Our primary motivation for developing a method for Bayesian inference is that obtaining the posterior on dependence structure is desirable over point estimates in applications such as moving object interaction analysis. That is, in Chapter 6 we make no assumptions as to the existence of a "true/correct" sequence of dependence structure and focus on characterizing posterior uncertainty in the relationships among multiple moving objects. We use a STIM for this task and specialize our discussion and notation in this section to such models. That is, we adopt a dynamic dependence model which uses TIM state conditional distributions specified by directed structures $\bar{E}^k$ and parameters Θ^k.

In the next section we define a tractable prior over the structure and parameters of a STIM. We then consider a Markov chain Monte Carlo (MCMC) approach for obtaining samples from the joint posterior rather than an exact/full characterization (cf. [1, 36]). We show that, while the joint samples of $z_{1:T}$ and $\mathbf{E}$ drawn using our MCMC approach

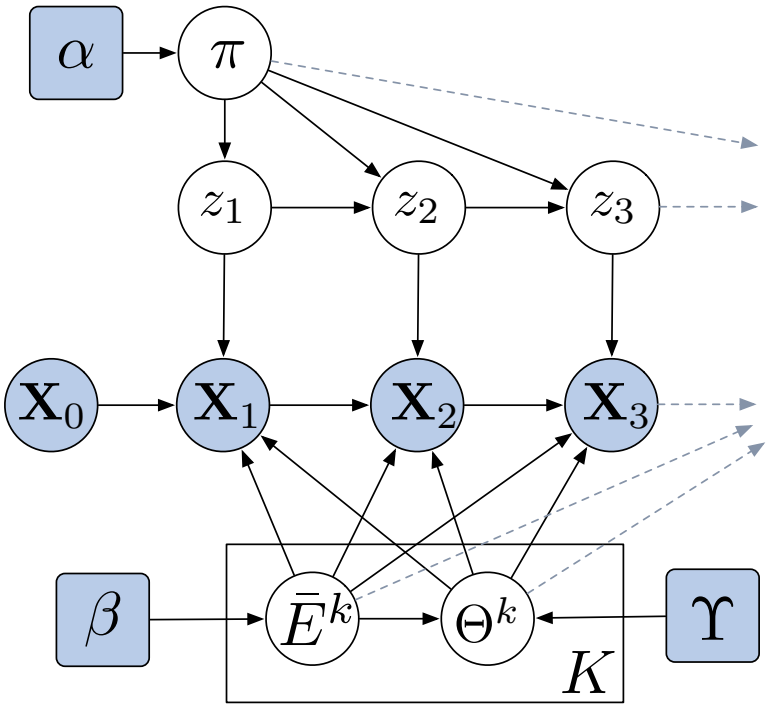


Figure 4.7. *A STIM(0,K) shown as a directed Bayesian network:* The latent state z_t indexes the directed structure $\bar{E}^k$ and parameters Θ^k of a TIM to describing the evolution of all time-series $\mathbf{X}_t$. Hyperparameters α specify a prior on the transition distribution described by π. Hyperparameters β and Υ are used to define priors on directed structure and parameters respectively.

are used as a proxy to the true posterior, given a specific set of labeled time points in which $z_t = k$ one can always obtain **exact** posteriors over structure $\bar{E}^k$ as discussed in Section 3.4.3.

■ 4.4.1 A Prior for STIMs

The generative model for a STIM(1,K) was introduced in Section 4.2. Here, we discuss placing priors on the parameters of this model. We assume the prior on all parameters and structures factorizes as:

$$p_0\left(\mathbf{E}, \mathbf{\Theta}\right) = p_0\left(\pi^0, \ldots, \pi^K\right) \prod_{k=1}^{K} p_0\left(E^k, \Theta^k\right). \tag{4.24}$$

Each state transition parameter set π^k is treated independently and a Dirichlet prior distribution is used (which is conjugate for a multinomial):

$$p_0(\pi) = \prod_{k=0}^{K} p_0\left(\pi^k | \alpha^k\right) \tag{4.25}$$

$$= \prod_{k=0}^{K} \mathrm{Dir}\left(\pi^k; \alpha_1^k, ..., \alpha_K^k\right). \tag{4.26}$$

Each hyperparameter α_j^k can be interpreted as a pseudo-count of how many times one has seen a transition from state k to state j. State persistence (approximating a semi-markov process) can be favored by setting

$$\alpha_j^k = c \;\forall\; j \neq k \quad \text{and} \quad \alpha_k^k = bc, \tag{4.27}$$

where c is a constant number of pseudo counts and $b > 1$ is a multiplier for c to favor self transitions.

The prior for the structure and parameters of the state conditional TIMs takes the form described in Section 3.4.2:

$$p_0\left(\bar{E}^k, \Theta^k\right) = p_0\left(E^k|\beta\right) p_0\left(\Theta^k | \bar{E}^k, \Upsilon\right) \tag{4.28}$$

An alternative model could place state specific priors with each state having its own β^k and Υ^k. Figure 4.7 depicts a STIM(0,K) along with its priors as a dynamic Bayesian network.

■ 4.4.2 MCMC Sampling for a STIM

Given observed time-series in $\mathcal{D}$ and a prior over parameters and structure of a STIM, we wish to characterize the posterior over structure at each point in time. Again, this is a difficult task due to the large space of K^T possible state sequences. Here, we take an MCMC approach. Iterative MCMC methods are a useful for drawing samples from an otherwise intractable target distribution. These methods construct a Markov chain that has the target distribution of interest as its equilibrium distribution. That is, after a large number of iterative steps a sample from this chain is a valid sample from the target distribution. The question of how many iterations are sufficient is a difficult one to answer and guarantees on obtaining valid samples are only asymptotic [1, 36]. In practice we tend to initialize multiple samplers and run them for a fixed period of

time, examining their output in the context of the problem of interest to determine if sufficient time was given.

Here, we design a *Gibbs sampler* for drawing samples from the posterior using a STIM. A Gibbs sampler is specific type of MCMC method that is well suited to our hidden state dynamic dependence model in that it uses exact conditional distributions that are tractable to compute using a STIM [37]. Our Gibbs sampler has three main steps which are outlined in Algorithm 4.4.1.

Algorithm 4.4.1 The Three Main Steps a STIM Gibbs Sampler.

Require: previous sample of the set of structures $\bar{\mathbf{E}}^{(i-1)}$ and parameters $\boldsymbol{\Theta}^{(i-1)}$

% The three main steps

Step 1. $z_{1:T}{}^{(i)} \sim p\left(z_{1:T}|\mathcal{D}, \bar{\mathbf{E}}^{(i-1)}, \boldsymbol{\Theta}^{(i-1)}\right)$

Step 2. $\{\pi^{0(i)}, \ldots, \pi^{K(i)}\} \sim p\left(\pi|z_{1:T}{}^{(i)}, \alpha\right)$

Step 3. $\{\bar{\mathbf{E}}^{(i)}, \boldsymbol{\Theta}^{(i)}\} \sim p\left(\bar{\mathbf{E}}, \boldsymbol{\Theta}|\mathcal{D}, z_{1:T}{}^{(i)}\right)$

The first step samples the state sequence given a previous sample of structures and parameters. This is done efficiently with backward message passing followed by forward sampling. High level details of Step 1 are shown in Algorithm 4.4.2. A simple modification of Step 1 is used to initialize the sampler by sampling a state sequence using only the transition prior rather than incorporating information from the data.

In Step 2, the sampled counts of state transitions are recorded. Let n_j^k for $k \in \{0, .., K\}$ and $j \in \{1, ..., K\}$ be a count of the number times $z_{t-1}{}^{(i)} = k$ and $z_t{}^{(i)} = j$. Given these quantities we then sample the transition probabilities using Algorithm 4.4.3.

In Step 3, a vector $\mathbf{t}_k$ is formed with all the time points with $z_t = k$. The structure and parameters are then sampled given $\mathcal{D}_{\mathbf{t}_k}$ for each k. See Algorithm 4.4.4. Step 3 requires one to sample from the posterior over structure and parameters. Details can be found in Appendix ??. It is important to note when using a switching temporal interaction model, one can efficiently calculate exact event probabilities and posterior over structures given a specific state sequence as described in Section 3.4.3. That is, one can perform static dependence analysis on the data $\mathcal{D}_{\mathbf{t}_k}$ for each k.

Algorithm 4.4.4 exposes how information about each parameter Θ^k and structure $\bar{E}^k$ is obtained from distinct parts of the data $\mathcal{D}_{\mathbf{t}_k}$. Thus, there is more information to take advantage of when determining the structure / state at each point in time, in contrast to a windowed approach. We refer the reader to Chapter 6 for illustrative

Algorithm 4.4.2 Step 1 of the STIM Gibbs Sampler During Iteration i

Require: The previous sampled structures $\bar{\mathbf{E}}^{(i-1)}$ and parameters $\mathbf{\Theta}^{(i-1)}$

% Generate backward messages $m_k^t = p\left(\mathcal{D}_{(t+1):T}|z_t = k, \widetilde{\mathcal{D}}_{t+1}, \bar{\mathbf{E}}^{(i-1)}, \mathbf{\Theta}^{(i-1)}\right)$
for $k = 1$ to K **do**
  $m_k^T \leftarrow 1$ **% Initialize Messages**
end for
for $k = 1$ to K **do**
  for $t = T - 1$ down to 1 **do**
    $m_k^t \leftarrow \sum_{j=1}^K \pi_j^{k(i-1)}\ p\left(\mathcal{D}_{t+1}|\widetilde{\mathcal{D}}_{t+1}, \bar{E}^{j(i-1)}, \Theta^{j(i-1)}\right) m_j^{t+1}$
  end for
end for

% Sample state sequence $z_{1:T}{}^{(i)}$ **working sequentially forward in time**
for $t = 1$ to T **do**
  for $k = 1$ to K **do**
    % Using probability $f_k = p\left(z_t = k|z_{1:(t-1)}, \mathcal{D}\right)$
    $f_k \leftarrow \pi_k^{z_{t-1}{}^{(i)}(i-1)} p\left(\mathcal{D}_t|\widetilde{\mathcal{D}}_t, \bar{E}^{k(i-1)}, \Theta^{k(i-1)}\right) m_k^{t+1}$
  end for
  % Sample the state assignment for $z_t{}^{(i)}$
  $z_t{}^{(i)} \sim \text{Discrete}\left(z_t; f\right)$
end for

Algorithm 4.4.3 Step 2 of the STIM Gibbs Sampler During Iteration i.

Require: n_j^k, for each $k \in \{0, .., K\}$ based on $z_{t-1}{}^{(i)} = k$

% Sample π^k
$\pi^{k(i)} \sim \text{Dir}\left(\pi^k; \alpha_1^k + n_1^k, ..., \alpha_K^k + n_K^k\right)$

Algorithm 4.4.4 Step 3 of the STIM Gibbs Sampler During Iteration i

Require: Sampled $z_{1:T}{}^{(i)}$, for each $k \in \{1, ..., K\}$

Let $\mathbf{t}_k = \{t\ |z_t{}^{(i)} = k\}$

% Calculate the posterior given data at points $\mathbf{t}_w$ **and sample**
$\left\{\bar{E}^{k(i)}, \Theta^{k(i)}\right\} \sim p\left(\bar{E}^k, \Theta^k|\mathcal{D}_{\mathbf{t}_w}, \widetilde{\mathcal{D}}_{\mathbf{t}_w}, \beta, \Upsilon\right)$

experiments and specific examples using this Gibbs sampler.

4.5 Related Work

As discussed in the sections above, our DDM is closely related to a standard HMM. A DDM differs in that the latent state indexes not only parameters, but structures as well. Specifically, the dependence structure among observations is a function of the *value* of the hidden state. A standard graphical model cannot capture the relationship between the *value* of a random variable and dependence structure. We hid this relationship in Figure 4.1 by representing all time series as a single random variable, $\mathbf{X}_t$. If one wishes to expose the details of how structure changes as a function the value of nodes in a graphical representation, a new set of semantics beyond that of standard graphical models is required.

The study of models that can encode structure as a function of the value of a random variable/node can be traced back to Heckerman and Geiger's similarity networks and Bayesian multinets [44, 35]. These models can represent *asymmetric independence* assertions, i.e. variables are independent for some but not all their values. Similarity networks are represented as a collection of Bayesian networks, each of which holds true for a particular setting of a hypothesis random variable. These networks were used primarily in expert diagnosis systems for various diseases. The representation allows for an expert on a particular disease to design a Bayesian network that only includes the variables/symptoms that are relevant for diagnosing that disease. They did not discuss inference in the context of these similarity networks. Their focus was primarily on providing a way to convert a similarity network to a full Bayesian network. The asymmetric independence assumptions were then encoded in the *parameters* of conditional probability distributions rather than in the topology of the network. Thus, a standard inference algorithm would not have access to the added independence information and could not take explicit advantage of it to reduce computation.

Bayesian multinets were introduced in an effort to enhance similarity networks and provide more efficient inference techniques. Multinets, much like similarity networks, are represented by a multiple Bayesian networks each centered around a single hypothesis variable/node. Each individual network holds true for a particular subset of hypotheses. However, unlike similarity networks, each network in a multinet contains all variables. Geiger [35] describes a general algorithm for efficient inference on multi-

nets and showed how the encoded asymmetric independence assumptions can reduce storage requirements and reduce computation. He also showed how this algorithm can be used for general inference on similarity networks by providing a way to convert a similarity network into a Bayesian multinet. Seeking even more flexible models Geiger and Heckerman also introduced the concept of a generalized similarity network in which there may be multiple hypothesis nodes with relationships between them.

This class of models was further explored and formalized by Boutilier, *et al.*'s Context-Specific Independence (CSI) [11]. CSI is a general notion that encompasses the idea of asymmetric independence discussed by Heckerman and Geiger. CSI describes situations in which two variables, X and Y, are independent given a certain context c, *i.e.* a particular setting of values to certain random variables. Boutilier, *et al.* showed how to verify global CSI statements given a set of local CSI statements in a network. Their work also showed how CSI yields regularities in the conditional probability tables (CPTs) for each variable in a Bayesian network. Unlike Geiger and Heckerman who used multiple networks to encode additional independence assumptions, [11] used a structured representation for their condition distributions to encode CSI. They focused on the use of decision-tree structured conditional probability tables (CPT) and show how this particular representation for CSI is compact and can be used to support effective inference algorithms. In a companion paper Friedman and Goldszmidt [32] showed how this structured representation of CPTs can aid in learning Bayesian networks from data.

More recently Milch, *et al.* have discussed further generalizations of Bayesian networks that focus on making CSI explicit [64]. They introduced the concept of partition-based models (PBMs) in which the conditional probability distribution of each variable is determined by a particular partitioning of the outcome space rather than on a set of parents. They provide a specific implementation of a PBM called a contingent Bayesian network (CBN). A CBN combines the use of structured CPTs (as in [11]) with labeling edges in a Bayesian network with contexts/conditions in which the edges are active/present. This edge labeling provides a simple representation in which CSI relationships can be more easily read from a graphical depiction of the model. A CBN can also contain cycles and have an unbounded number of latent variables. Milch *et al.* discuss the conditions in which a CBN defines a unique distribution and provide an algorithm for approximate inference.

In [7], Bilmes introduced a model for time-series that makes use of CSI. His dynamic

Bayesian multinet is a hidden state model in which the value of the hidden state at time t encodes the dependence structure among the observations within a local time window surrounding t. Bilmes' work focused on techniques for learning the state conditional dependence structure from labeled training data to help improve classification. These techniques use a greedy pairwise criteria for adding new edges to encode causal relationships in a network. He focused on the problem domain of speech recognition and demonstrated how learning class-specific (state-specific) discriminative structure can help improve performance.

Our DDM belongs to this general class of models as do the related models of [7, 52, 94]. It is most closely related to Bilmes' dynamic Bayesian multinet in that when using a TIM we are reasoning over directed causal structures. However, our inference task is that of unsupervised discovery of structure rather than learning a better model for predictive analysis given training data. In addition, by using the static dependence models and inference tools discussed in Chapter 3 we are able to provide exact Bayesian reasoning over structure. That is, we do not rely on greedy search or other approximate methods for structural discovery.

4.6 Summary

In this chapter, the static dependence models discussed in Chapter 3 were extended to allow structure to change over time. This was done by introducing the class of DDMs. A DDM uses a dynamically evolving latent variable to index both structure and parameters of an underlying static dependence mode over time. We discussed how dynamic dependence analysis can be performed via inference on a DDM and contrasted this approach with standard windowed analysis. In contrast to ML inference on static dependence model, we showed how ML inference on a DDM can take advantage of parametric differences when distinguishing structure. This analysis was supported by a set of illustrative experiments. By placing priors on the transition distribution of the DDM along with those presented in Chapter 3 for a TIM, we demonstrated how one can using a Gibbs sampler for Bayesian inference. While this approach only produces samples from the posterior, given a sampled state sequence *exact* marginal posterior statistics can be calculated.

Chapter 5

Application: Audio-Visual Association

As technology rapidly advances, electronics used for digital audio-visual recording and storage have become less expensive and their presence in our environment more ubiquitous. In addition to movies, television shows and news broadcasts, inexpensive recording devices and storage have allowed for business meetings, judicial proceedings and other less formal group meetings to be digitally archived. At the same time, technologies such as automatic speech recognition [60] and face detection and recognition [91, 89, 95] have allowed for useful metadata to be extracted from these archives. Searches can then be performed to quickly find videos which contain certain phrases or involve particular individuals. In this chapter, we explore the problem of determining which, if any, of the individuals in the video are speaking at each point in time. Whereas one could make use of strong prior models of audio-visual appearance *when* they are available, here we show that the problem can be cast as one of audio-visual association, demonstrating the utility of the previous development and bypassing the need for strong prior assumptions. Specifically, we frame the problem in terms of dynamic dependence analysis.

Hershey and Movellan have previously showed how measuring correlation between audio and video pixels can help detect who is speaking in a scene [46]. Nock and Iyengar [68] provided an empirical study of this technique on a standardized dataset, CUAVE [72]. Further study of detecting and characterizing the dependence between audio and video was carried out by Slaney and Covell [84] and Fisher *et al.* [30]. All of these techniques process data using a sliding window over time assuming a single audio source within the analysis window. As such, they do not incorporate data outside of the analysis window. In this chapter, we show empirically that by treating the problem as a dynamic dependence analysis task in which data over all time is considered, one

can exploit audio and visual appearance changes associated with who is speaking. In doing so, we achieve the best performance to date on a standard dataset for audio-visual speaker association. This performance was achieved without any window parameters to set, without a silence detector or lip tracker and without any labeled training data.

We begin by introducing the three datasets used in our experiments. We then briefly discuss the basic audio-visual features extracted for input to our dynamic dependence analysis in Section 5.2. In Section 5.3 we map reasoning over audio-visual association to reasoning over the dependence structures of a static dependence model. Lastly, we present results in Section 5.4.

■ 5.1 Datasets

For our audio-visual speaker association experiments we use three separate datasets. Each involves recordings of a pair of individuals taking turns speaking. All datasets contain video recorded at 29.97 frames per second. We convert each video frame to a 256 value grayscale image. The audio for each dataset was resampled at 16kHz and broken up into segments aligned with the video frames.

■ 5.1.1 CUAVE

The CUAVE dataset [72] is a multiple speaker, audio-visual corpus of spoken connected digits. We use the 22 clips from the *groups* set in which two speakers take turns reading digit strings and then proceed to utter two different strings simultaneously. In order to compare to prior work [68, 40], we only consider the section of alternating speech. In each clip both individuals face the camera at all times. Ground truth was provided by Besson and Monaci [6].

A separate video stream is extracted for each speaker in the CUAVE corpus. This is done using a face detector [91] and simple correlation tracking of the nose region in order to obtain a stabilized face. Figure 5.1.1 shows a sample frame from raw video in addition to extracted faces for each individual in the dataset. As seen in the Figure 5.1.1, the CUAVE dataset, has fairly high-resolution frames with an uncluttered background. As such empirical results are useful for relative comparisons rather then absolute performance.

(a) Sample frame from one sequence in the CUAVE dataset

(b) Extracted Faces for all CUAVE sequences

Figure 5.1. *The CUAVE Dataset:* a) A sample video frame from one sequence. b) Individual frames from the automatically extracted video streams for each face for all 22 sequences. Faces were extracted using a face detector and simple correlation tracking to stabilize the region around a person's nose.

■ 5.1.2 Look Who's Talking

While the CUAVE corpus provides a standardized dataset for comparing performance with prior work, the individuals recorded do not interact with each other and are always facing the camera. Our second dataset is a single video recorded in the same style as the CUAVE database in which two individuals take turns speaking digits. However, while the speaker looks into the camera, the non-speaking turns to look at the speaker.

(a) Sample Frame (b) Extracted Faces

Figure 5.2. *"Look Who's Talking" Data:* a) A sample video frame b) The extracted hand-labeled face regions.

This provides a dataset in which there is a strong appearance change associated with who is speaking, as may be the case in a meeting where participants look toward the current speaker.

A separate video stream is extracted for each individual. In this video the individuals change their head pose throughout the video. One could develop an extension to the simple face tracker used for the CUAVE dataset and attempt to extract a representation for each person's face which is stabilized and normalized for head pose. However, such an approach would throw out potentially informative features. That is, in this video head pose is an excellent cue as to who is speaking. Ideally, one would extract the head pose information and uses them as additional observed features. Here, we take the simple approach of hand labeling a rectangular region in the video for each person and crop the video accordingly. That is, we sacrifice perfectly tracked faces for a simple representation that can still capture strong appearance changes that occur with a change of speaker. Figure 5.1.2 shows a sample frame from the raw video and extracted face regions.

■ 5.1.3 NIST Data

Lastly, we move away from scripted speech scenarios and explore more realistic data involving meeting conversations. The third dataset is a single camera recording of a sequence from the NIST meeting room pilot corpus [34] (sequence 20011115-1050

(a) Sample Frame (b) Extracted Faces

Figure 5.3. *NIST Data:* a) A sample video frame b) The extracted face regions using face detection information provided to the VACE program.

camera 1). This data was provided as part of the Advanced Research Development Agency (ARDA) Video Analysis and Content Exploitation (VACE) program.

This sequence is a recording of four individuals taking turns speaking. Face location information was provided by participants in the VACE program. We took the supplied face track information and extracted a video sequence for two individuals facing the camera in the sequence being analyzed. Figure 5.1.3 shows a sample frame and the extracted faces. Note that in this dataset the extracted face regions have poor resolution relative to the previous datasets.

■ 5.2 Features

While there has been much work in exploring and learning informative features for audio-visual association tasks [68, 40, 46, 84, 30], the focus of this dissertation is on performing dynamic dependence analysis given a pre-specified set of features. To this end, we choose video and audio features that capture the basic dynamics of each modality as well as their static appearance.

We use frame-based features for the video streams extracted for each individual in scene as well as the audio. That is, a set of features is extracted for each frame of video. There are two basic types of features for each time-series. Static features for frame t are calculated based on the data within that frame. Dynamic features at frame

t incorporate information from neighboring frames.

5.2.1 Video Features

For each sequence there are two video time-series; one for each individual in the scene. Given the raw grayscale video sequence for each individual we extract a simple set of static appearance features for each frame. These static features are calculated by by first performing principal component analysis (PCA) [74] (c.f. [50]) on the entire sequence and extracting the top 40 principal components. On average using 40 components captured over 90% of the energy for each sequence. The video sequence is then transformed by projecting each frame onto this 40 component basis, yielding 40 coefficient values for each frame of video.

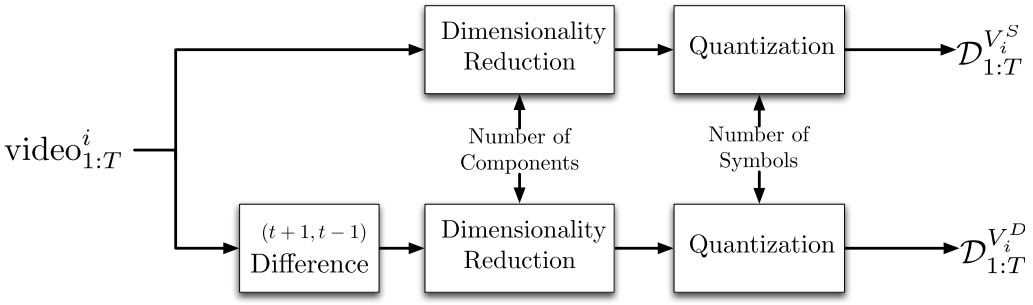
We calculate dynamic features by first taking raw pixel-wise frame differences. At frame t, a difference image was formed from the raw frame at time $t+1$ and $t-1$. This yields a difference video for each sequence. Similarly to the static features, PCA is performed on this video sequence and 40 coefficients for each frame are then extracted.

Each of these feature streams is then vector-quantized. A 20 symbol codebook is learned separately for the dynamic and static features by learning a 20 component Gaussian mixture model using EM. Each stream is then encoded using its corresponding codebook, yielding a discrete observation for each stream at each frame [1]. Figure 5.4(a) summarizes how the video features are obtained via a block diagram.

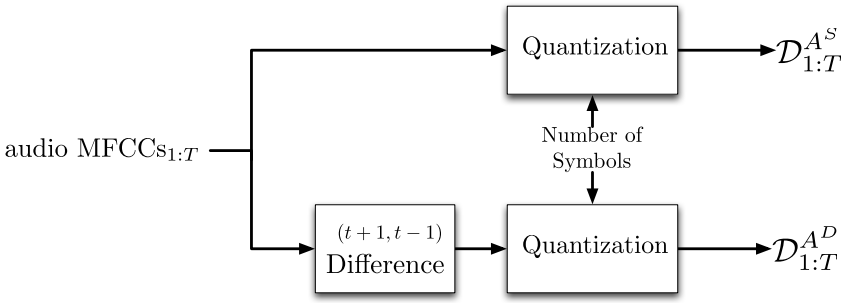
5.2.2 Audio Features

For each sequence analyzed there is a single audio stream broken up into segments corresponding to video frames. We calculate 13 Mel-frequency cepstral coefficients (MFCCs) for each of these frames [63, 13] to form what we will refer to as our static audio features. MFCCs are common perceptually motivated feature used in automatic speech recognition. Dynamic features at frame t are, again, formed by taking the difference between the raw static MFCC features at frame $t+1$ and $t-1$. These audio feature streams are separately vector quantized using learned 20 symbol codebooks. Figure 5.4(b) summarizes how the video features are obtained via a block diagram.

[1] We arbitrarily chose 20 symbols at first. Some quick analysis of smaller and larger codebooks showed little change in performance. However, more thorough analysis must be done before drawing any conclusions about codebook size. Here, we simply provide a fixed video extract technique for all datasets.



(a) Video Processing



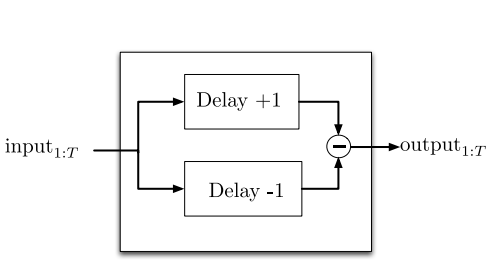
(b) Audio Processing

Figure 5.4. *Video and Audio Feature Extraction Block Diagrams:* Sub-blocks are explained in Figure 5.5. (a) A block diagram summarizing how both the static $\mathcal{D}^{V_i^S}$ and dynamic $\mathcal{D}^{V_i^D}$ video features for person i are obtained. (b) A block diagram summarizing how the static $\mathcal{D}^{A^S}$ and dynamic $\mathcal{D}^{A^D}$ audio features are obtained.

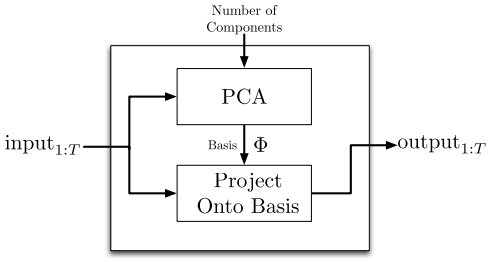
■ 5.2.3 Inputs to Dynamic Dependence Analysis

For each sequence analyzed, this yields three quantized static feature time-series; one representing the audio, $\mathbf{x}^{A^S}_{1:T}$, and two representing the video for each person in the scene, $\mathbf{x}^{{V_1}^S}_{1:T}$ and $\mathbf{x}^{{V_2}^S}_{1:T}$. In addition, there are three corresponding dynamic feature streams, $\mathbf{x}^{A^D}_{1:T}$, $\mathbf{x}^{{V_1}^D}_{1:T}$ and $\mathbf{x}^{{V_2}^D}_{1:T}$. These are the $N = 6$ time-series input for analysis. For clarity, we will index them as $A^S, A^D, {V_1}^S, {V_1}^D, {V_2}^S$ and ${V_2}^D$, rather than 1 through 6, where the superscript denotes (S)tatic or (D)ynamic features and the subscript indexes the speaker.

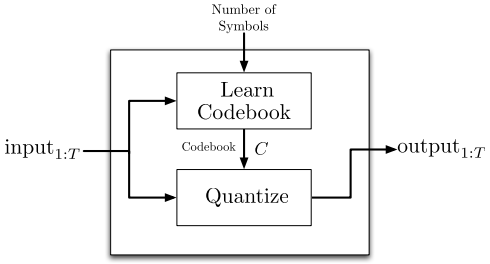
It is important to note that the dimensionality reduction with PCA and codebook learning is done **separately** for each stream and for each data sequence analyzed. That is, no user or dataset specific training is being performed.



(a) $(t+1, t-1)$ Difference



(b) Dimensionality Reduction



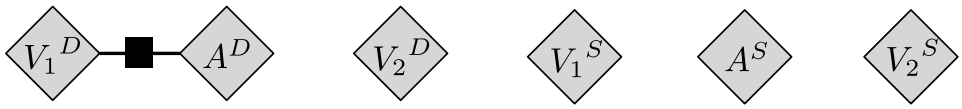
(c) Quantization

Figure 5.5. *Sub-blocks for Audio and Video Processing:* (a) A diagram showing how the the $(t+1, t-1)$ sub-block works. It outputs the difference between a positive and negative temporal delay of its input. (b) A diagram showing how dimensionality reduction is performed. Treating each sample independently PCA is performed on all of the input data to learn a basis Φ with a specified number of components. All the data is then projected onto this basis. (c) A diagram showing how vector quantization is performed. All the data input data is used to learn a codebook with a specified number of symbols. We form this codebook by learning a Gaussian mixture model. The data is then quantized using this codebook, encoding each sample with the index of the mixture component best describes the sample.

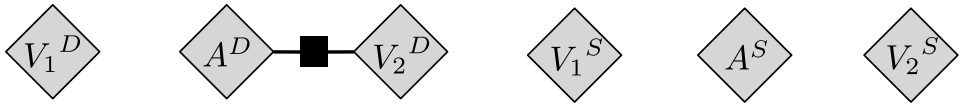
■ 5.3 Association via Factorizations

Our goal is to determine which, if any, of the people in a given sequence are producing the recorded audio at each point in time. We map this problem to that of determining which video observation is associated with the audio observation. We describe this association in terms of a factorization on the input time-series. For this task we define a finite set of factorizations we wish to identify.

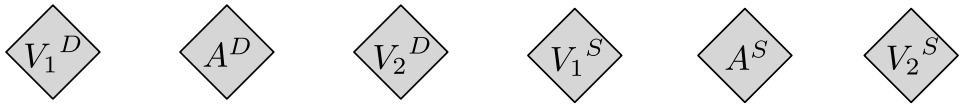
When individual 1 is speaking we expect the audio to be associated with the video for individual 1. Similarly, we expect the audio to be associated with the video individual 2 when her or she is speaking. When neither of the individuals are speaking we expect no association among the input time-series. Thus, we consider three possible factorizations



(a) F^1 : Person 1 is speaking



(b) F^2 : Person 2 is speaking



(c) F^3: Neither person is speaking

Figure 5.6. *The Three Audio and Video Factorizations Considered:* The factorizations are shown as association graphs. When person 1 is speaking we assume the feature time-series factorize as shown in (a). When person 2 is speaking we assume (b). When neither person is speaking we assume all observed time-series are independent.

being active at any point in time:

$$F^1 = \{\{A^D, {V_1}^D\}, \{{V_2}^D\}, \{A^S\}, \{{V_1}^S\}, \{{V_2}^S\}\} \tag{5.1}$$

$$F^2 = \{\{A^D, {V_2}^D\}, \{{V_1}^D\}, \{A^S\}, \{{V_1}^S\}, \{{V_2}^S\}\} \tag{5.2}$$

$$F^3 = \{\{A^D\}, \{{V_1}^D\}, \{{V_2}^D\}, \{A^S\}, \{{V_1}^S\}, \{{V_2}^S\}\} \tag{5.3}$$

Figure 5.6 shows association graphs for these three factorizations. Note that the structural differences between these 3 factorizations are only in the dynamic features. The implicit assumption is that the dependence information is primarily capture by the dynamics of the audio-visual speech process while static features primarily exhibit appearance changes (i.e. parametric) rather than dependence changes. As discussed in Chapter 4 while a windowed approach for analysis will only be able to take advantage of the structural differences, a dynamic dependence model can use these static features to help distinguish which is the correct factorization.

■ 5.4 AV Association Experimental Results

We perform dynamic dependence analysis on these datasets using three different approaches. The first approach performs windowed analysis comparing FactMs using the three possible factorizations defined in Equations 5.1, 5.2 and 5.3. We use the abbreviation WFT to denote this as a windowed factorization test. For the WFT, at each frame, the likelihood ratios $\hat{l}_{1,2}$, $\hat{l}_{1,3}$ and $\hat{l}_{2,3}$ are calculated using a window of samples centered around that frame. Note that, by virtue of an additional edge, both F^1 and F^2 are more expressive than the fully independent F^3. Thus, additionally, p-values for $\hat{l}_{1,3}$ and $\hat{l}_{2,3}$ are calculated via 100 permutations. If $\hat{l}_{1,2}$ is positive (negative) then we eliminate F^2 (F^1) as a possible hypothesis and use the p-value for $\hat{l}_{1,3}$ ($\hat{l}_{2,3}$) to choose between F^1 (F^2) and F^3. That is, we first compare which video stream is most likely associated with the audio by looking at $\hat{l}_{1,2}$. Given the most likely of the two we then look at the p-value for that hypothesis when compared to no association, F^3, to decide if we should reject F^3. Window sizes of 8, 15, 30, 60, 90 and 120 frames with various p-value thresholds are tested. For each sequence, the final result recorded is the one with the window size and p-value threshold that gives the best performance for that particular sequence. While selection of the settings which yield the best performance is not reasonable in practice, our purpose is to use these results as a best-case baseline for comparison to the dynamic approach.

The second approach uses an HFactMM with(0,3) with each state linked to a single factorization. That is, state i indexes the factorization F^i allowing the semantic label of "Who is speaking" to be directly read from the state sequence inferred by the model. Similarly, the third approach uses a FactMM(0,3) which removes the Markov structure on the state sequence. We perform ML inference using both of these models to get the most likely state sequence after learning parameters via EM. For EM, 100 random initializations are used with a maximum of 80 iterations each. The most likely parameters are kept. In most cases EM converges before 40 iterations and the maximum is found by multiple initializations.

■ 5.4.1 CUAVE

All three approaches were applied to all 22 sequences in the CUAVE corpus. Table 5.1 shows a summary of the average performance for each method. Performance is reported in terms of the percentage of frames correctly classified according to the ground truth

	HFactMM	FactMM	Best WFT	Nock and Iyengar[68]	Gurban and Thiran[40]
Mean Accuracy (%)	80.24	78.51	83.86	75	87.4
Mean Accuracy C(%)	**88.11**	86.38	83.42	NA	NA

Table 5.1. *Results Summary for CUAVE:* The Best WFT accuracy corresponds to the WFT with settings that maximized the average performance for the entire dataset. C= silence constraint imposed

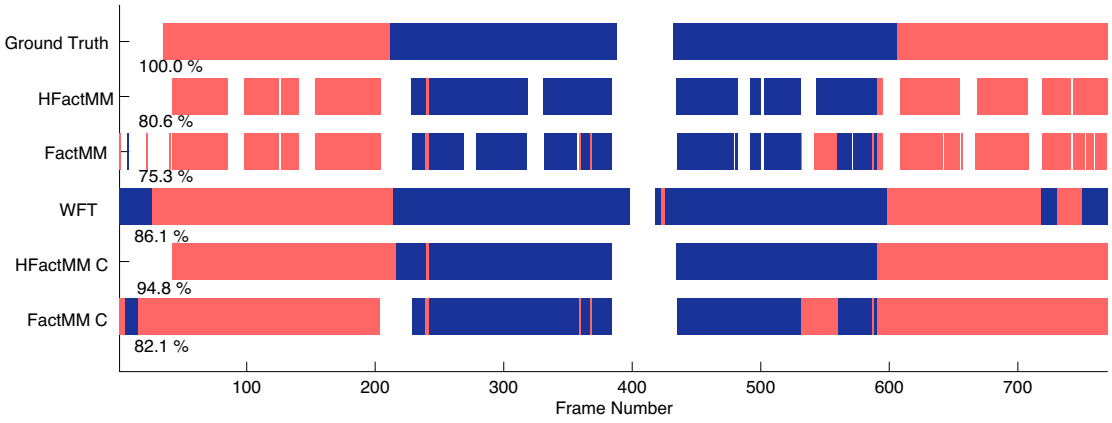


Figure 5.7. *Results for CUAVE Sequence g09:* The top row indicates the ground truth labeling of who is speaking. White = neither person is speaking, light red = person 1, dark blue = person 2. The following rows display the output of dynamic dependence analysis using an HFactMM, a FactMM, and a windowed approach (WFT). The bottom two rows show post processed results for the HFactMM and FactMM in which short (< 25 frames) segments of silence/neither person speaking are removed to be consistent with the ground truth labeling procedure. The accuracy of each result is shown below the labeling.

provided by [6]. The first row of Table 5.1 shows that all the techniques give around 80% accuracy. Typically, the maximum average performance of the WFT was obtained with a window length of 30 frames, approximately one second. This shows that, with some training data to set window length and p-value thresholds, the WFT method would do well with these features. However, these results are somewhat misleading, as we will explain.

Figure 5.7 shows the estimated labels for a typical sequence in the corpus (g09). The top line shows the ground truth labeling. The next two are the outputs of the HFactMM

and FactMM. Notice that these methods disagree with the ground truth by consistently putting non-speaking (fully independent) blocks between speaker transitions and within speaking blocks. Examination of these sections in the original video reveals that they are actually short periods of silence. Consequently, the HFactMM and FactMM correctly labeled these sections. The WFT does not exhibit this behavior and smoothes over the short silence regions.

This disagreement with the provided ground truth is an artifact of how the ground truth was defined. In [6], it is stated that:

> A group of frames is labeled as silent when it is composed of at least 25 frames.

That is, periods of silence less than 25 frames within speech are considered part of the speech according to the ground truth. While one might question this constraint, we can easily incorporate it into our analysis to gain a fairer comparison. We impose the constraint by post processing the outputs to remove any periods of labeled silence ($z_t = 3$) less than 25 frames. The constrained outputs are shown in the last two lines of Figure 5.7. With this constraint the HFactMM and FactMM outperform all other methods, improving to 88% and 86% respectively as shown in Table 5.1. To the best of our knowledge these results are equivalent to or better than all other reported results for audio-visual speaker association on the CUAVE group set. Nock and Iyengar [68] report 75% accuracy with a windowed Gaussian MI measure and Gurban and Thiran [40] get 87.4% with a *trained* audio-visual speech detector. Both methods use a silence/speech detector and only perform a dependence test when there is speech. Our method obtains better performance without the benefit of separate training data or a silence detector.

Full detailed results for the CUAVE Corpus are found in Table 5.2.

■ 5.4.2 Look Who's Talking Sequence

In the CUAVE database, much of the information about who is speaking comes from the changes in dependence structure between the audio and the video. This is evident in the fact that a windowed approach performs reasonably well, yielding 83% accuracy. Appearances changes in the CUAVE dataset are primarily a result of differences in the voice characteristics of each individual and the fact that each individual only moves

	HFactMM	FactMM	HFactMM C	FactMM C	WFT 8	WFT 15	WFT 30	WFT 60	WFT 90	WFT 120
g01	84.77	80.19	**94.61**	93.67	69.54	77.49	80.73	80.19	80.19	80.19
g02	84.42	80.65	**92.09**	89.57	65.08	80.28	78.27	71.48	74.50	74.37
g03	87.37	78.26	**92.75**	93.89	67.49	79.92	80.43	74.53	75.78	72.05
g04	88.06	86.91	**96.06**	93.63	73.00	87.60	88.88	90.15	87.14	86.67
g05	87.58	83.30	**95.38**	90.99	59.67	68.57	76.70	82.20	83.41	76.04
g06	68.49	65.76	68.17	68.33	71.22	79.58	84.41	**85.21**	82.15	76.21
g07	83.78	82.16	**96.49**	93.78	70.81	83.24	89.73	90.54	89.19	92.16
g08	80.95	75.73	**92.46**	84.82	71.28	77.56	80.56	82.40	82.79	80.66
g09	80.62	75.28	**94.78**	82.11	74.16	88.32	86.09	77.52	75.90	81.86
g10	78.14	72.78	80.41	74.02	69.28	81.03	80.41	**84.12**	81.44	80.62
g11	74.76	79.47	83.80	83.99	83.24	84.56	89.83	**93.41**	89.27	86.63
g12	61.91	64.20	69.85	72.81	59.08	67.70	**78.47**	75.10	71.33	70.79
g13	80.31	76.31	**88.85**	86.24	72.82	83.28	86.59	83.97	85.37	84.32
g14	83.25	78.93	**93.19**	82.59	71.34	79.84	80.50	80.50	79.97	80.24
g15	85.92	84.02	**93.11**	90.47	70.82	80.35	82.70	81.23	85.34	82.11
g16	69.86	77.40	71.78	84.52	70.96	85.75	**85.89**	85.21	83.97	82.74
g17	85.36	81.66	93.65	93.12	90.48	90.12	**95.41**	93.65	90.83	88.01
g18	64.23	82.37	72.06	87.84	51.86	59.28	69.48	72.16	77.53	74.64
g19	89.16	83.05	85.34	**85.80**	58.32	75.27	77.86	76.95	71.30	68.85
g20	84.81	80.11	**97.01**	90.80	78.07	90.16	91.23	88.56	86.20	85.13
g21	82.58	81.52	**92.42**	88.70	77.39	85.90	85.51	84.18	80.98	83.11
g22	78.81	77.13	**94.21**	88.57	80.18	85.21	88.26	89.79	86.74	83.08
Mean	80.24	78.51	**88.11**	86.38	70.73	80.50	83.54	82.87	81.88	80.48

Table 5.2. *Full Results on CUAVE Group Dataset:* C=silence constraint imposed outputs (non-speaking segments shorter than 25 frames removed) WFT # indicates a windowed factorization test with window length #. Note that the WFT results are obtained by finding a p-value threshold to give the maximum mean accuracy over the entire dataset. The best result for each sequence is highlighted in **bold**.

his or her mouth when speaking[2]. In our second dataset there is also a significant appearance change when there is a transition of who is speaking. When one person is speaking the other subject changes his or her gaze. The results for this sequence are shown in Figure 5.8. Both the HFactMM and FactMM greatly outperform the WFT as predicted in our theoretical analysis. The poor results of the WFT show that there is not sufficient dependence information in the features at all times. However, the HFactMM and FactMM take advantage of the static appearance differences (in this case head pose) to help group the data and correctly label the video.

■ 5.4.3 NIST Sequence

Lastly, we explore performance on the NIST data. Note that, in this dataset, when neither tracked person is speaking there is often an individual off-camera who is speaking rather than simple silence. Consequently, the independent factorization, F^3, must

[2]Note that this is not entirely true. Most people are continuously moving their mouth in some way while anticipating his or her turn to speak in the CUAVE dataset

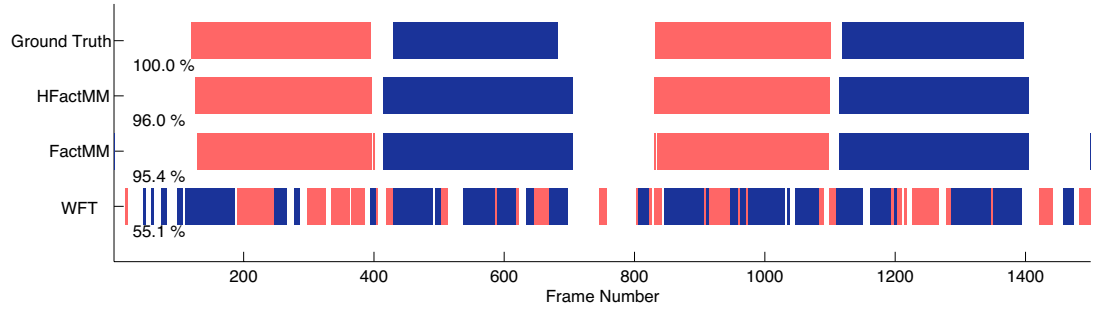


Figure 5.8. *Results for the "Look Who's Talking" Data:* White = neither person is speaking, light red = person 1, dark blue = person 2. The top row shows the ground truth. Results are shown for the HFactMM, a FactMM and windowed test. The accuracy of each result is shown below the labeling. Note here the HFactMM and FactMM do significantly better than the windowed approach. This is due to the fact they can cluster data from far way time-points when making a local decision.

model more than just silence. Figure 5.9 shows the results of all three approaches when applied to this data. The HFactMM again achieves the best performance, but only obtains 76% accuracy when compared to the ground truth. We see that the best choice of parameters for the WFT favor the independent factorization. This, again, indicates there is little dependence information in the feature and much is gained by using a dynamic model.

This data also reveals that the problem of audio-visual association is much more challenging than the controlled CUAVE dataset indicates. Note that we purposely choose simple features and focused on demonstrating the advantage of using a dynamic dependence model when treating this problem as a generic dynamic dependence analysis task. We expect one can improve performance by incorporating better face tracking and robust audio and video features.

■ 5.4.4 Testing Underlying Assumptions

In the previous sections we explored the use of a dynamic dependence model on three audio-visual speaker association datasets. We focused on comparing the results obtained using an HFactMM with those obtained using a windowed approach. Performance was discussed in the context of the overall "difficulty" of each dataset. Difficulty was loosely defined in terms of how well each technique performed, the type of speech activity and the amount audio-visual appearance change associated with changes of speaker in the data. Here, we provide some additional analysis. We explore how well the underlying

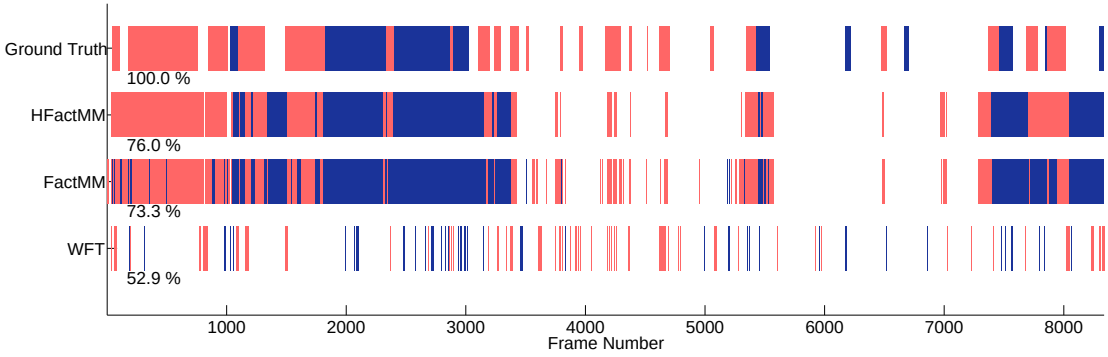


Figure 5.9. *Results for NIST data:* White = neither person is speaking, light red = person 1, dark blue = person 2. The top row shows the ground truth. Results are shown for the HFactMM, a FactMM and windowed test. The accuracy of each result is shown below the labeling. Again, here the HFactMM and FactMM significantly out perform the windowed approach, which sees very little dependence in the data. The dynamic models overcome this by pooling data over all time.

assumptions made when mapping the audio-visual association problem to a dynamic dependence analysis task hold for each dataset.

The main underlying assumption we make in treating the speaker association problem as a dynamic dependence analysis task is that dependence structure among the observations is tied to who is speaking. That is, when person i is speaking we assume video i is dependent on the audio and independent of all other observations. We also assume there is no dependence among the observations when none of the observed individuals are speaking.

We test this assumption on all datasets by measuring the mutual information between the audio and video for each person over various window sizes. That is we measure $I(A; V_i)$ for each $i \in \{1, 2\}$ within a specified window of time. As discussed in Chapter 3, mutual information is a natural choice of statistic to measure due to its relationship to the likelihood of association under the factorization model . Here, we simplify our notation such that the audio, A, within a window $\mathbf{t}_w$, is the set of both static and dynamic audio features, $A_{\mathbf{t}_w} = \left\{\mathcal{D}^{A^s}_{\mathbf{t}_w}, \mathcal{D}^{A^d}_{\mathbf{t}_w}\right\}$, and similarly for video $V_{i,\mathbf{t}_w} = \left\{\mathcal{D}^{V_i^s}_{\mathbf{t}_w}, \mathcal{D}^{V_i^d}_{\mathbf{t}_w}\right\}$. Recall that these audio and video features were computed over the entire sequence. In particular, this means that the codebook has to represent both speakers in the case of the audio measurements.

Figure 5.4.4 shows a summary of this analysis for a single sequence in the CUAVE

dataset. The summary is broken down into three rows of plots, each of which corresponds the state of who is speaking. Each plot summarizes the measured mutual information for each person, $I(A;V_1)$ and $I(A;V_2)$, over various window lengths given a particular state. The x axis indicates the window length. For each window length we plot the mean and standard deviation of mutual information measured within all windows with that specified length as a bar graph with error bars. Each plot also shows two horizontal lines, one for each person, which display the mutual information measured using all of the data within the specified state.

Examining Figure 5.4.4 we see that, for this CUAVE sequence, when neither person is speaking our assumptions hold and there is no dependence among the audio and video of either person overall all such times or within windows. The second and third rows show that $I(A;V_i)$ is highest when person i is speaking. However, they also show that there is some measured dependence between the audio and the person who is not speaking. While within a window this measured dependence may be spurious or a result of using a small amount data, we see that when using all the data (the horizontal lines) the dependence is still present. The results for different window lengths show that variance of measured dependence decreases with window length and that smaller window lengths tend to have higher mutual information. These results also show that it is the difference in dependence that matters, not the presence of dependence, in this sequence.

Figure 5.4.4 shows results for the "Look Who's Talking" data. We see trends similar to that of the CUAVE data. However, the there is some measured dependence when neither person is speaking, the measured dependence is lower over all states, and the difference in mutual information for each speaker is less. That is, the analysis seems to indicate there is less dependence information in this data. Results for the NIST data are shown in Figure 5.4.4. We see that while the average dependence within windows for all states is generally higher than the other two sets of data, there is more variance and the mutual information measured using all the data within a state is actually lower than that in CUAVE and the "Look Who's Talking" data for each state. Such characteristics can help explain why we obtained slightly worse performance on the NIST sequence relative to the others.

Next we explore how well we can distinguish the state of who is speaking from the overall distribution of the data. While the previous analysis looked at dependence information, here we wish to analyze both parametric and dependence differences among the

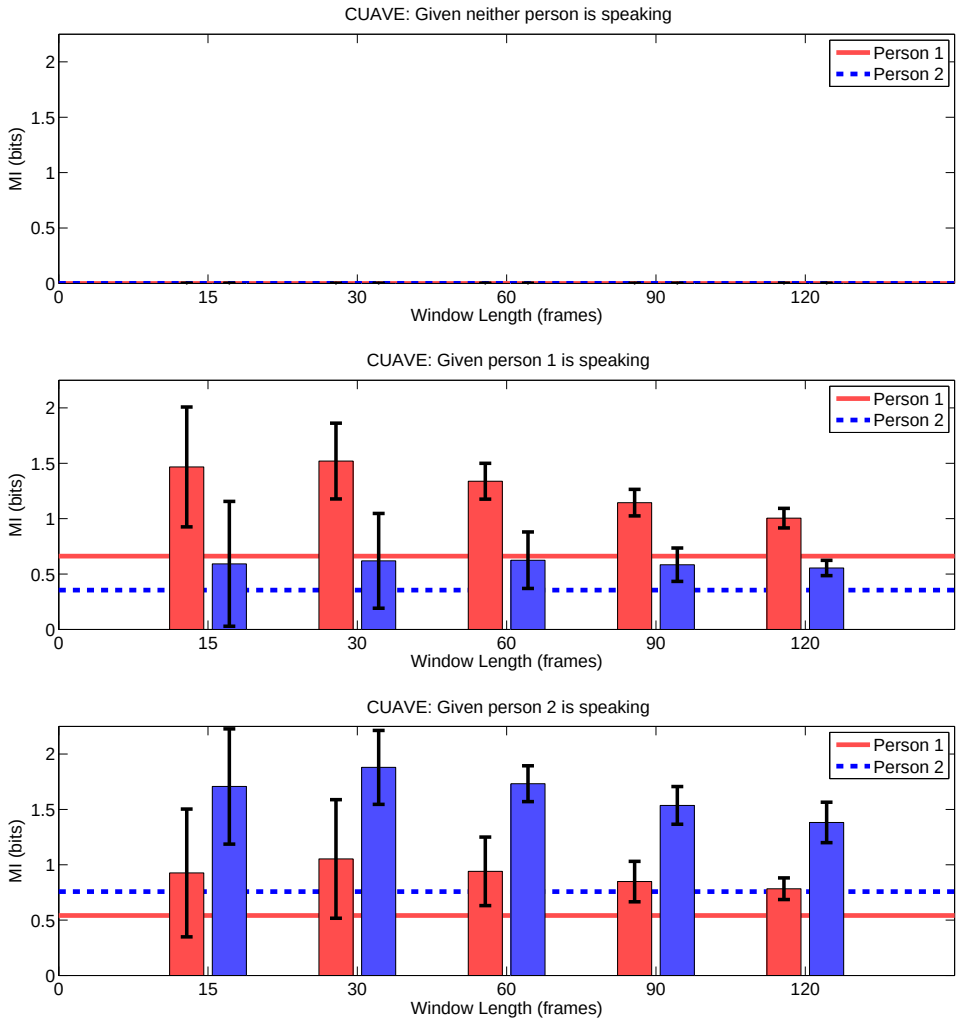


Figure 5.10. *Dependence Analysis for CUAVE Sequence g09:* Each row is a separate state. The x axis is the window length used. For each window length two bar graphs show the average and standard deviation of $I(A;V_1)$ (red) and $I(A;V_2)$ (blue) measured using all windows of the specified length within the specified state. The horizontal lines show $I(A;V_1)$ (red) and $I(A;V_2)$ measured using all the data within the given state.

distribution of observations for each state. In our model we assume that distribution on observations changes depending who is speaking. We test this assumption by measuring the mutual information of the state z with the data $\mathcal{D}$. We also look at $I(z;A)$, $I(z;V_1)$ and $I(z;V_2)$ to help understand which observations are most informative about the state. Figures 5.13(a), 5.13(b), and 5.13(c) shows results of this analysis for each

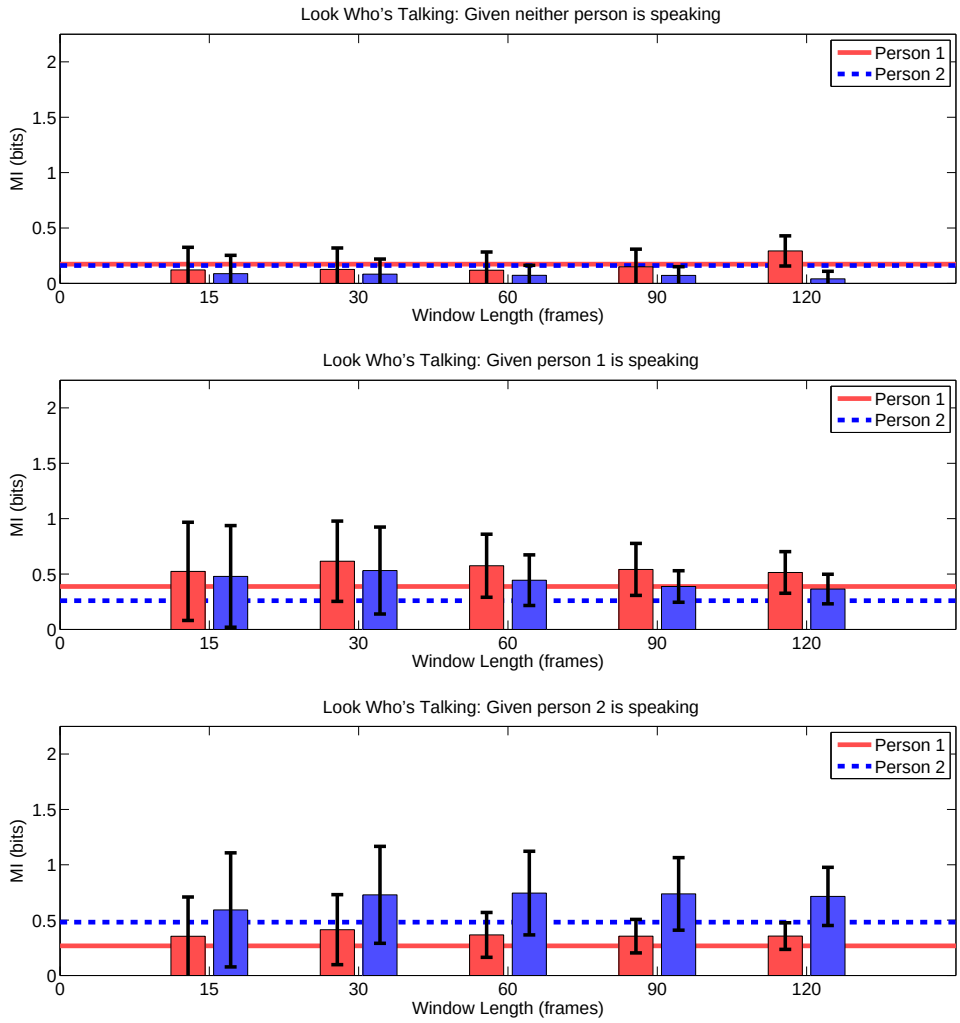


Figure 5.11. *Dependence Analysis for the "Look Who's Talking" Sequence:* Each row is a separate state. The x axis is the window length used. For each window length two bar graphs show the average and standard deviation of $I(A;V_1)$ (red) and $I(A;V_2)$ (blue) measured using all windows of the specified length within the specified state. The horizontal lines show $I(A;V_1)$ (red) and $I(A;V_2)$ measured using all the data within the given state.

dataset. The bars display the measured mutual information, while the horizontal line is the entropy of the state label $H(z)$ which serves as an upper bound. Figure 5.13(a) shows that for a CUAVE sequence the mutual information is close to the upper bound and that most of information comes from the video observations rather than the audio. Figure 5.13(b) shows a similar trend for the "Look Who's Talking dataset". However,

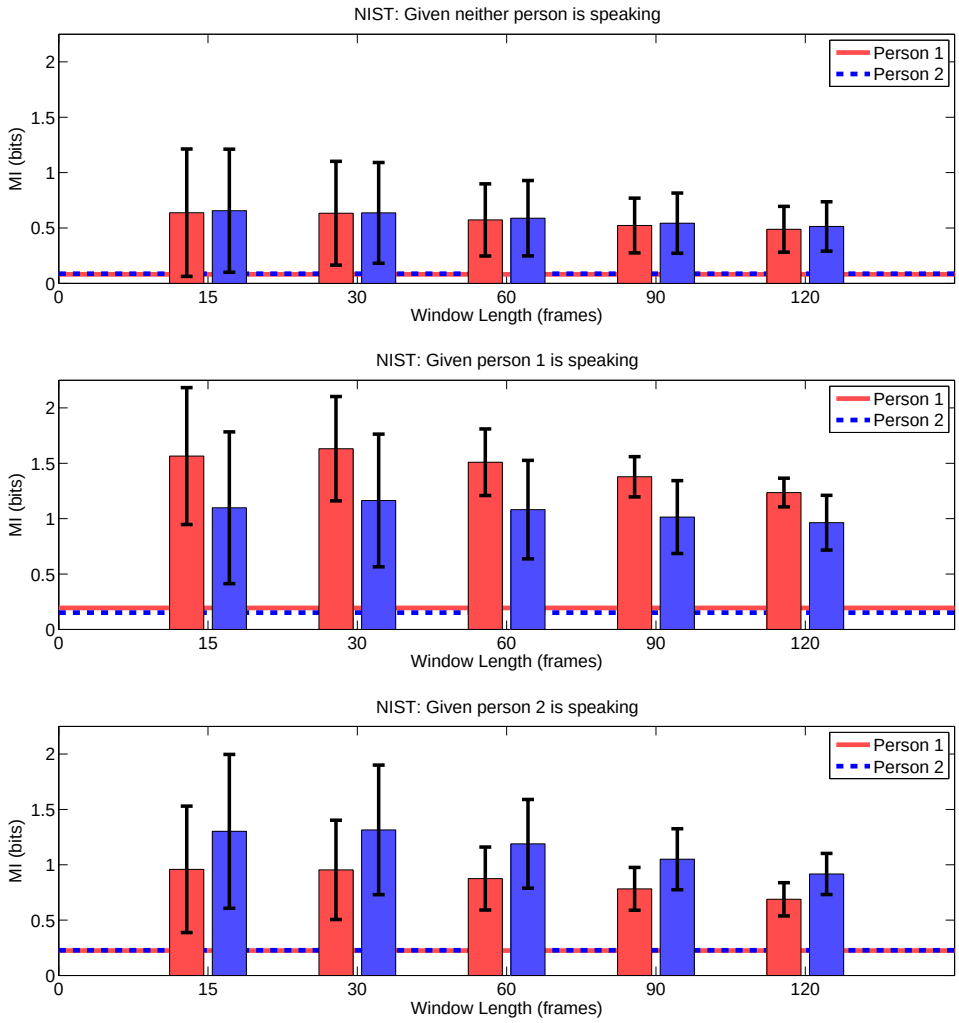
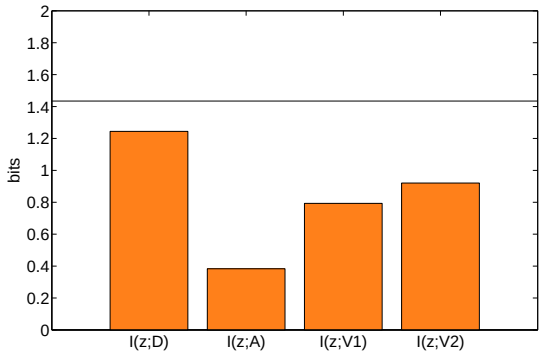
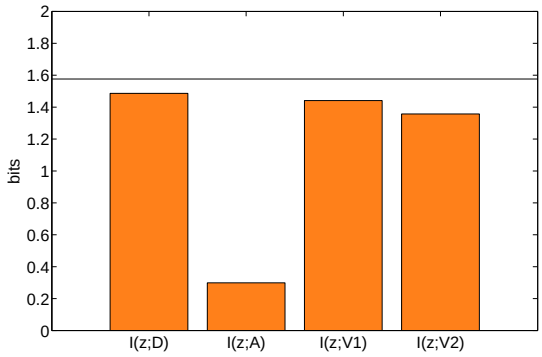


Figure 5.12. *Dependence Analysis for the NIST Sequence:* Each row is a separate state. The x axis is the window length used. For each window length two bar graphs show the average and standard deviation of $I(A;V_1)$ (red) and $I(A;V_2)$ (blue) measured using all windows of the specified length within the specified state. The horizontal lines show $I(A;V_1)$ (red) and $I(A;V_2)$ measured using all the data within the given state.

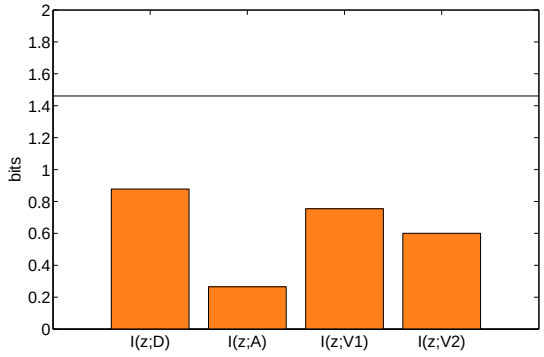
the mutual information is higher and closer to the upper bound. Lastly, Figure 5.13(c) shows that for the NIST data the state is less distinguishable than in the other datasets. However, it is important to again note that this analysis is dependent on our feature choices. Perhaps features other than MFCCs might result in stronger dependence in the audio features.



(a) CUAVE g09



(b) Look Who's Talking



(c) NIST

Figure 5.13. *Analysis of State Distinguishability:* Each subfigure shows the analysis for a different dataset. The bars show the mutual information between the state label z and all observations $\mathcal{D}$ or just the audio A or video V_1/V_2. The horizontal line is the entropy of the state label $H(z)$ which is an upper-bound on the mutual information

In our experiments, we use an HFactMM with a single state for each dependence structure, implicitly assuming the model is stationary when that state is active. The windowed method is largely insensitive to the model varying so long as the dependence is measurable. Next, we test this stationarity assumption by looking at two different windows of data that both occur within a specified state. We measure the mutual information between the label of which window the data came from and the data. Low mutual information indicates that the statistics are similar in both windows and indicate our assumption of stationarity may be true. Figures 5.14, 5.15, 5.16 show our analysis results. Note that the upper bound on mutual information is 1 bit assuming each window is equally likely. We plot the average mutual information and variance as bar plots for multiple window lengths for each state. We break up the analysis into comparing two windows that occur within the same utterance (the state label does not change any time between when the windows occur) versus those which occur in different utterances.

Figures 5.14, 5.15, 5.16 show that, over all datasets and states, within utterance stationarity improves with larger window sizes. This trend is most evident in the CUAVE and "Look Who's Talking" data, and much less in the NIST data. That is, even within the same utterance the distribution varies a lot in the NIST data. Windows with different utterances seem to be very distinct in general. This is somewhat unexpected, as one would expect more of a downward trend with larger window lengths. Even though the utterances are different one would expect the visual appearance and audio characteristics would be similar.

Reviewing all the analysis presented in this section we now summarize our findings. Our assumptions about changing dependence structure more or less hold in that the video for the speaker tends to be most dependent on the audio. This was first strongly hinted at by the fact we achieved state-of-the-art performance on each of these datasets. Ranking how closely each dataset agrees with our assumption they can be ordered as: CUAVE, "Look Who's Talking", NIST. Second, our assumption of each state having a distinct distribution seems to hold true as well. Ranking the datasets based on this assumption we order them as: "Look Who's Talking", CUAVE, NIST. This is consistent with fact that we achieved a large improvement in performance by using an HFactMM rather than a windowed approach on the "Look Who's Talking" data. Lastly, our assumption of stationarity seems to be violated in each data sequence we explored. This raises the question of whether this is a poor assumption or that the

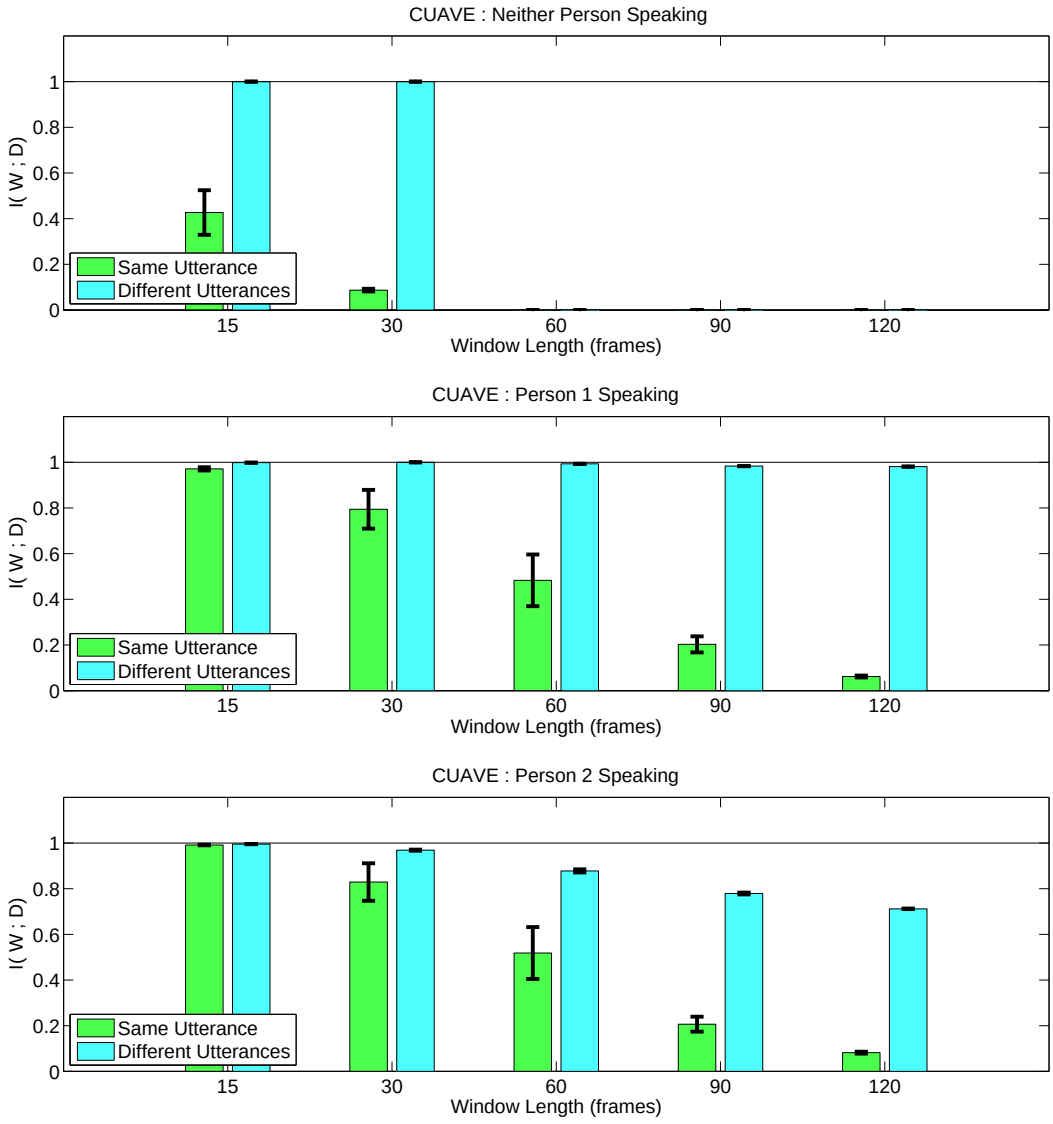


Figure 5.14. *Stationarity Analysis for CUAVE Sequence g09:* Each row is a separate state. The x axis is the window length used. For each window length, two windows are extracted within the specified state. For each pair of windows the mutual information $I(W;\mathcal{D})$ in bits is measured where W is the label indicating which window the data came from. We summarize the results for each window in two bar plots. The first (green) gives the mean and variance of the mutual information when the windows are within the same utterance. The second (light blue) is the same for when the windows are in different utterances. The blank bars for window lengths above 60 frames in the first plot are due to lack of data meeting those criteria.

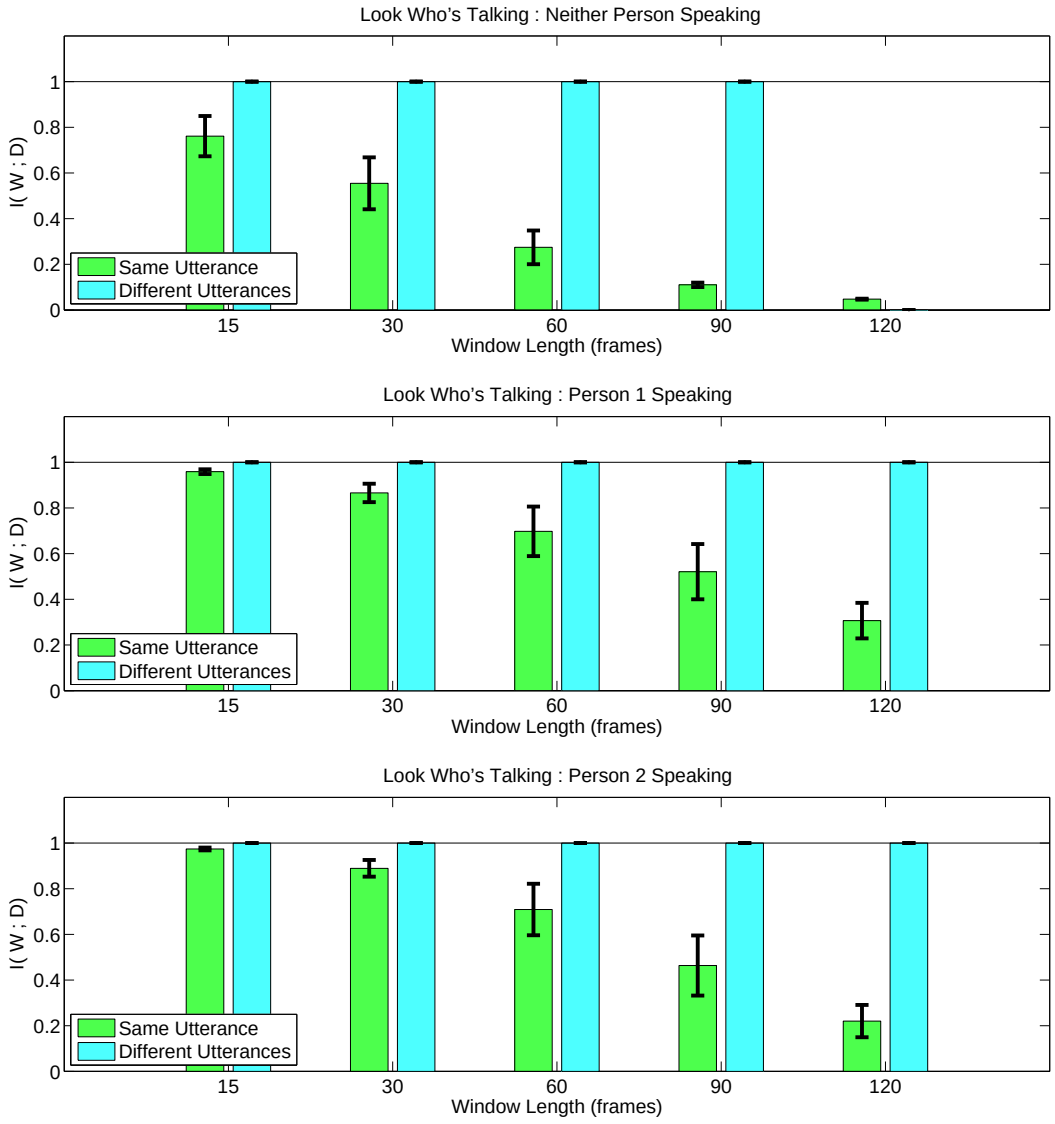


Figure 5.15. *Stationarity Analysis for the "Look Who's Talking" Sequence:* Each row is a separate state. The x axis is the window length used. For each window length, two windows are extracted within the specified state. For each pair of windows the mutual information $I(W;\mathcal{D})$ in bits is measured where W is the label indicating which window the data came from. We summarize the results for each window in two bar plots. The first (green) gives the mean and variance of the mutual information when the windows are within the same utterance. The second (light blue) is the same for when the windows are in different utterances. The blank bar for different utterances of window length 120 is due to lack of data meeting those criteria.

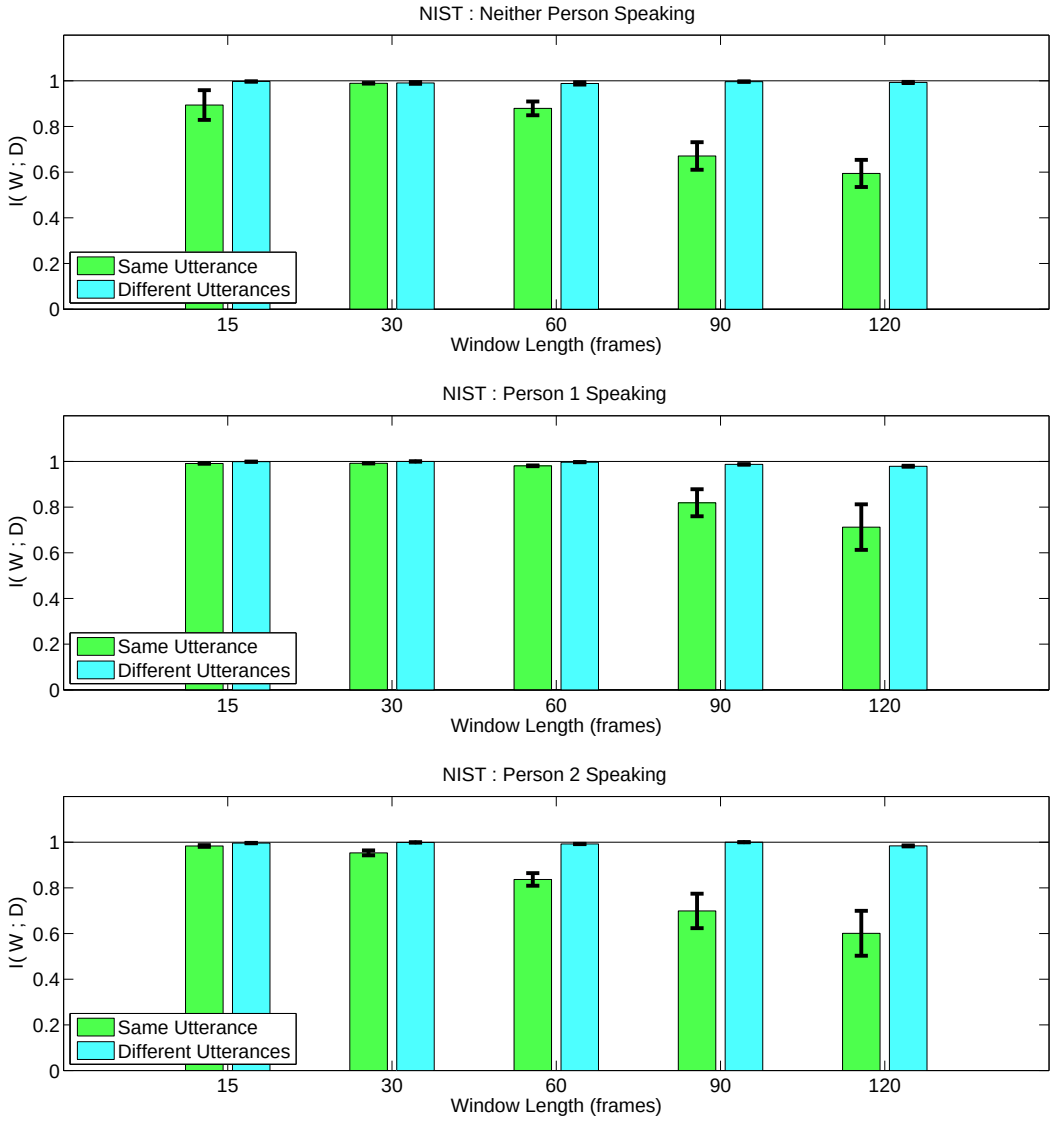


Figure 5.16. *Stationarity Analysis for the NIST Sequence:* Each row is a separate state. The x axis is the window length used. For each window length, two windows are extracted within the specified state. For each pair of windows the mutual information $I(W;\mathcal{D})$ in bits is measured where W is the label indicating which window the data came from. We summarize the results for each window in two bar plots. The first (green) gives the mean and variance of the mutual information when the windows are within the same utterance. The second (light blue) is the same for when the windows are in different utterances.

lack of stationarity is a result of the features we choose or particular data we looked at. Either way, it suggests some future avenues to explore when analyzing and/or extending the models proposed in this dissertation.

■ 5.5 Summary

In this chapter, we looked at the problem audio-visual speaker association. We cast the problem in terms of dynamic dependence analysis using an HFactMM to reason over factorizations. Consistent with the theoretical analysis provided in Chapter 4, we have empirically shown that by modeling dependence as a dynamic process the HFactMM can exploit both structural and parameter differences to distinguish between hypothesized states of dependence. This is in contrast to sliding window methods which can only discriminate based on structural differences. State-of-the-art performance was obtained on a standard dataset for audio-visual association. Significantly, this was achieved without benefit of training data.

Chapter 6

Application: Object Interaction Analysis

In this chapter, we use dynamic dependence models for the purpose of analyzing the interactions among multiple moving objects. Such analysis can provide information and insight into coupled behavior for automated surveillance and tracking of multiple objects [85, 20]. Much of the past work in this area has focused on modeling and classifying object activity [39, 2, 14], specific single object behaviors [15, 9, 71] and anomalous events [12, 66]. In contrast, we focus on the *unsupervised* identification of interactions among *multiple* moving objects.

Our approach is closely related to the recent work of Tieu [88] who has formulated the problem of detecting and describing interaction explicitly in terms of statistical dependence and model selection. That is, interactions are defined in terms of dependence structure rather than specific behavior models learned from training data. Representing object trajectories as time-series, [88] assumes a single dependence structure among them over all time. The strength of interactions are described in terms of statistical measures of dependence. While similar to this preliminary proof of concept, our approach differs in that, here, we are primarily concerned with characterizing a full posterior distribution over interactions described by the structure of a graphical model. More importantly, using the dynamic dependence models presented in Chapter 4, we efficiently reason over a larger set of interaction structures which are allowed to evolve over time.

We begin in Section 6.1 by casting the problem of reasoning over object interactions as one of Bayesian inference over the structure of a temporal interaction model (TIM). We discuss how to parameterize a TIM and provide an illustrative example using data generated from the model. A series of experiments on various datasets of moving objects

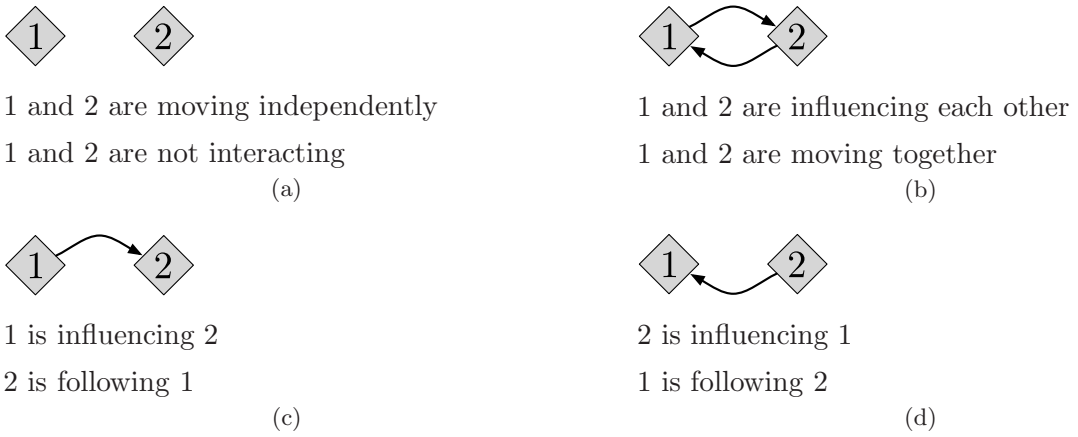


Figure 6.1. *Interaction Graphs for Two Objects:* Four possible directed interaction graphs describing the behavior of two objects. Under each graph are two descriptions of object interactions that would yield the structure above.

are conducted and analyzed in Section 6.2.

■ 6.1 Modeling Object Interaction with a TIM(r)

Consider N objects moving within an environment. We assume we are given information jointly about each object's location over a regularly sampled time interval, 1 to T. Time indexes the trajectory of the i-th object as time series $\mathbf{x}^i_{1:T}$, which describes the kinematic state[1] of that object at each point in time. We map the problem of identifying interactions among these objects to a dynamic dependence analysis task using these observed time-series. Specifically, we use an, r-th order temporal interaction model, TIM(r), whose directed structure $\bar{E}$ describes the causal relationships among the positions of the N objects under analysis.

For example, consider the case in which there are only two objects. Figure 6.1 shows the four possible direct structures $\bar{E}$ depicted as interaction graphs. Underneath each interaction graph are two possible semantic descriptions of interactions that may yield the depicted structure above. However, one must be careful to not invert the relationship and claim a semantic description can be read from the interaction graph alone. While generic terms such as "influence" may be appropriately read from the interaction graph, other terms such as "follow" cannot. That is, a directed edge from

[1] In our experiments, kinematic state is simply the 2-D position of the object.

1 to 2 indicates that the observation for object 2 at the current time is dependent on or "influenced" by the past of object 1. Terms such as "follow" are more complex and involve describing the nature of this particular influence. That is, the label "2 follows 1" is usually interpreted to mean object 2 is physically behind object 1. Such a label cannot be read from the dependence graph in Figure 6.1(c). The structure depicted can also be associated with an interaction involving "2 stays ahead of 1". The missing information telling us which description is more appropriate is captured by the value of the parameters associated with the edge from 1 to 2 (which we treat as a nuisance).

In other words, $\bar{E}$, as depicted in the interaction graph, only describes the structure of the interaction, while the parameters associated with that structure describe the nature of the interaction. In this chapter, we are primarily concerned with quantifying uncertainty in the structure of the interaction given observed data. However, as discussed in Chapter 4, the parameters describing to nature of object interactions will help to pool information when reasoning over interactions that may change over time. If interested in characterizing the nature of interaction, priors can be chosen for the parameters which favor certain behaviors and then the posterior on these parameters can be calculated given the observed data as discussed in Section 3.4.3.

■ 6.1.1 Parameterization

Before we can use a TIM(r) to infer interactions, we need to supply its parameterization. That is, we need to choose a parametric form for $p\left(\mathbf{X}_t|\widetilde{\mathbf{X}}_t, \bar{E}, \Theta\right)$ that can describe trajectories of objects. Recall from Section 3.4 that a TIM factorizes as

$$p\left(\mathbf{X}_t|\widetilde{\mathbf{X}}_t, \bar{E}, \Theta\right) = \prod_{v=1}^{N} p\left(\mathbf{x}_t^v|\widetilde{\mathbf{x}}_t^{v,\mathbf{pa}(v)}, \Theta_{v|\mathbf{pa}(v)}\right). \tag{6.1}$$

Thus, ultimately we must describe the form of $p\left(\mathbf{x}_t^v|\widetilde{\mathbf{x}}_t^{v,\mathbf{S}}, \Theta_{v|\mathbf{S}}\right)$ that models the dependence of a time-series v given its past in addition to the past of a parent set $\mathbf{S}$.

We adopt a simple r-th order vector auto-regressive model, VAR(r), for these conditional relationships [70]. While other models are possible, this particular choice is useful as it leads to a tractable conjugate prior. That is, assuming $\mathbf{x}_t^i$ describes the d-dimensional position of object i at time t ($\mathbf{x}_t^i \in R^d$), we adopt the following linear model

$$\mathbf{x}_t^v = A_{v|\mathbf{S}}\ \widetilde{\mathbf{x}}_t^{v,\mathbf{S}} + w, \tag{6.2}$$

where $A_{v|\mathbf{S}}$ is a $d \times d(1+|\mathbf{S}|)r$ dynamic matrix and w is zero-mean Gaussian noise with $d \times d$ covariance $\Lambda_{v|\mathbf{S}}$. The parameter $\Theta_{v|\mathbf{S}} = \{A_{v|\mathbf{S}}, \Lambda_{v|\mathbf{S}}\}$. Recall that $\widetilde{\mathbf{x}}_t^{v,\mathbf{S}}$ contains the past r values of time-series v in addition to the past values of all time-series in set $\mathbf{S}$. The VAR(r) model describes the current value of time-series v as a linear function of these past values plus Gaussian noise. Here, the parameter $A_{v|\mathbf{S}}$ captures the nature or form of dependence.

Consider a simple case with $r = 1$ in which time-series v has single parent u. In this situation, Equation 6.2 is

$$\mathbf{x}_t^v = \begin{bmatrix} A_{v,v} & A_{v,u} \end{bmatrix} \begin{bmatrix} \mathbf{x}_{t-1}^v \\ \mathbf{x}_{t-1}^u \end{bmatrix} + w \tag{6.3}$$

which is a simple VAR(1) model. Higher order models ($r > 1$) can be used to capture higher dependence on motion. For example a VAR(2) can be used to capture dependence on velocity when $\mathbf{x}_t^v$ represents position.

In this chapter, we will focus on Bayesian inference of structure while integrating out parameters. We wish to define prior on these parameters so that one can efficiently integrate over them.. Equation 6.2 is a linear regression model. As mentioned in Chapter 2 a conjugate prior for the parameters of this model is the matrix-normal-inverse-Wishart distribution:

$$\begin{aligned} p_0\left(\Theta_{v|\mathbf{S}}\right) &= p_0\left(A_{v|\mathbf{S}}|\Lambda_{v|\mathbf{S}}\right) p_0\left(\Lambda_{v|\mathbf{S}}\right) \\ &= \mathcal{MN}\left(A_{v|\mathbf{S}}; \Omega_{v|\mathbf{S}}, \Lambda_{v|\mathbf{S}}, \mathbf{K}_{v|\mathbf{S}}\right) \mathcal{IW}\left(\Lambda_{v|\mathbf{S}}; \Xi_{v|\mathbf{S}}, \nu_{v|\mathbf{S}}\right) \end{aligned} \tag{6.4}$$

Thus, posterior updates given observed data $\mathcal{D}$ can be performed using Equation 2.53, and the evidence $W_{\mathbf{S},v}$ can be calculated using the Matrix-T distribution given in Equation 2.59. This is done by mapping $\mathcal{D}^y = \mathcal{D}^v$, $\mathcal{D}^x = \widetilde{\mathcal{D}}^{v,\mathbf{S}}$ and the hyper-parameters $\Omega, \mathbf{K}, \Lambda$, and ν to their $v|\mathbf{S}$ indexed versions in Equation 6.4.

While all of our experiments will involve one or two-dimensional position information, it is straightforward to extend our approach to higher dimensional data.

■ 6.1.2 Illustrative Synthetic Example of Two Objects Interacting

In this section, we perform an illustrative experiment on two ($N = 2$) one-dimensional time-series. We wish to examine the degree to which one can distinguish whether a certain dependence relationship is present as a function of the number of samples

and strength of that dependence. We design a TIM(1) with a static structure of $\mathbf{x}_\mathbf{t}^1$ influencing $\mathbf{x}_\mathbf{t}^2$. That is, $\bar{E}$ contains a single edge, $1 \rightarrow 2$, as depicted in Figure 6.1(c).

The time-series $\mathbf{x}_\mathbf{t}^1$ is set to be a random walk such that $A_{1|\emptyset} = 1$ and $\Lambda_{1|\emptyset} = 1$. That is,

$$\mathbf{x}_t^1 = \mathbf{x}_{t-1}^1 + w^1 \tag{6.5}$$

where w^1 is zero mean, unit variance Gaussian noise. The amount of influence $\mathbf{x}_\mathbf{t}^1$ has on $\mathbf{x}_\mathbf{t}^2$ is controlled via a variable ρ such that $A_{2|1} = \begin{bmatrix}(1-\rho) & \rho\end{bmatrix}$ and $\Lambda_{2|1} = 1$. That is,

$$\mathbf{x}_t^2 = (1-\rho)\mathbf{x}_{t-1}^2 + \rho\mathbf{x}_{t-1}^1 + w^2 \tag{6.6}$$

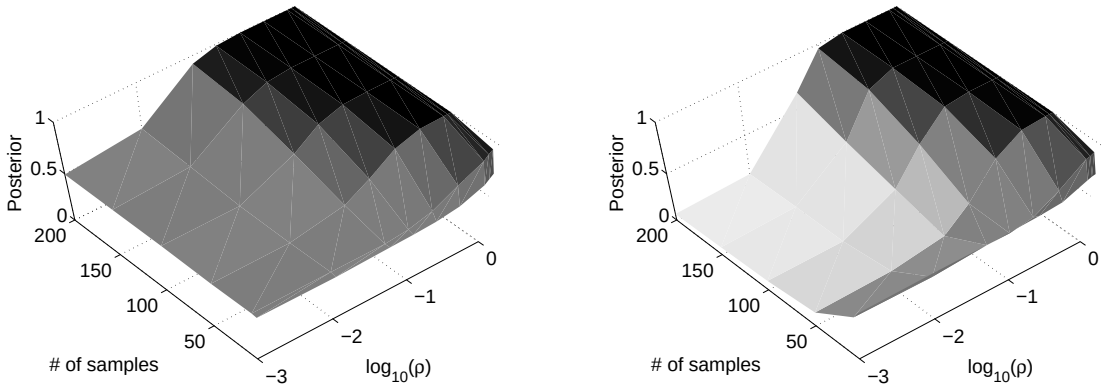
where w^2 is unit variance. Note that if $\rho = 1$, $\mathbf{x}_t^2$ is the position $\mathbf{x}_{t-1}^1$ plus noise, and if $\rho = 0$ it simply follows its own random walk independent of the other time-series.

Using a weak matrix-normal-inverse-Wishart prior on the parameters[2] and a uniform prior on structure (all β set to 1), we calculated the the posterior probability of edge $1 \rightarrow 2$ given a set of samples from our model. This is done for various settings of ρ and T. For each setting, 100 trials are performed and the average posterior edge probabilities are recorded. Note that, for $N = 2$, the set of all structures $\mathcal{A}_2$ (set of all structures) $= \mathcal{P}_2^1$ (set of single parent structures) and the number of structures in each set is: $|\mathcal{T}_2| = 2$ (directed trees), $|\mathcal{F}_2| = 3$ (directed forests) and $|\mathcal{A}_2| = 4$ (all directed structures).

Figure 6.2(a) and Figure 6.2(b) show results for structure sets $\mathcal{T}_2$ and $\mathcal{A}_2$ respectively. The results follow one's intuition: few samples or $\rho < 0.1$ results in an edge posterior close to chance, i.e. a uniform posterior over structure (1/2 for $\mathcal{T}_2$, 1/4 for $\mathcal{A}_2$). Once $\rho > 0.1$ there is a sharp increase in the posterior on the the edge being present.

Note that, here, we are quantifying uncertainty in terms of posterior edge appearance probabilities. As discussed in Section 3.4.3 the posterior probability of edge appearance is the same as the posterior mean of a multiplicative indicator function f on that edge,

[2]Throughout this chapter, when we use the term "weak" prior for matrix-normal-inverse-wishart distribution we generally set the prior to have the following parameters for all conditional relationships $v|\mathbf{S}$: The matrix-normal is set to be zero mean $\Omega = 0$ and $\mathbf{K}$ set to be the identity matrix. For the inverse-wishart we set the degrees of freedom ν to be the dimension of the data plus 2. Ξ is either set to be the identity for synthetic data or $2\hat{\Lambda}$ where $\hat{\Lambda}$ is the ML estimate of the covariance given *all* of the data being analyzed. These priors are "weak" in the sense that they do not heavily bias the posterior and the data term easily dominates the prior.



(a) $\mathcal{T}_2$ (b) $\mathcal{A}_2 = \mathcal{P}_2^1$

Figure 6.2. *The Average Edge Posterior for Different ρ and # of samples T:* The average was taken over 100 trials for each setting of ρ and T. (a) shows results when we restrict $\bar{E} \in \mathcal{T}_2$ (b) shows results when there are no restrictions placed on $\bar{E}$.

$\mathbb{E}_{p(\bar{E}|\mathcal{D})}\left[f(\bar{E})\right]$. One can also calculate higher-order information such as the posterior variance using $\mathbb{E}_{p(\bar{E}|\mathcal{D})}\left[f(\bar{E})^2\right]$. For example, running a single trial with $T = 50, \rho = 0.1$ one obtains a posterior edge probability of .6327 and posterior variance of .2324 when considering $\mathcal{T}_2$.

■ 6.2 Experiments

In this section we present a series of experiments on datasets involving the interaction of multiple, $N > 2$, tracked moving objects, At first, we look at a simple static dependence analysis task and then explore situations in which interactions are changing over time. Specifically, we are interested in quantifying uncertainty in the dependence structure describing interactions rather than obtaining point estimates. When analyzing data we do not assume there is a "true or correct structure", rather, our goal is to fully characterize the uncertainty over the structure of interactions.

■ 6.2.1 Static Interactions

We begin by considering a dataset comprised of recordings of three individuals playing a simple interactive computer game. This "Interaction Game" is similar to that used by Tieu [88]. Multiple players are each given a screen to look at which displays multiple

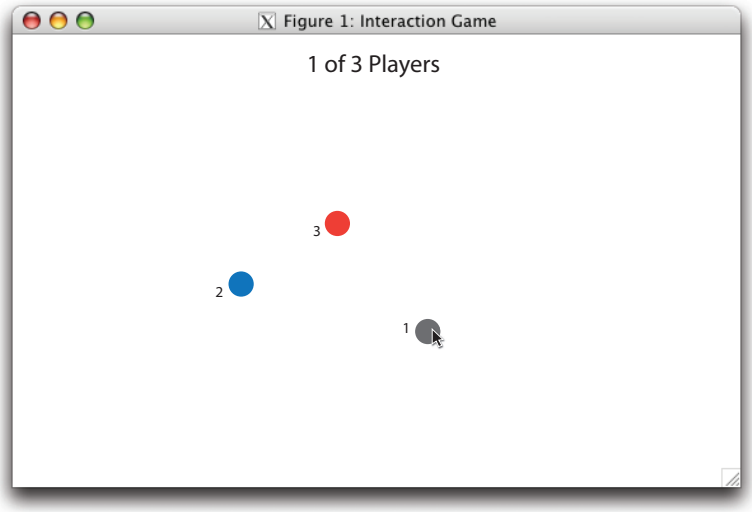


Figure 6.3. *A screenshot from our "Interaction Game":* Each player is represented by a round marker. Here, we see the interface from the point of view of player 1.

numbered markers. Each player's computer mouse controls the position of one specific marker on their screen while the other markers represent the positions of the other players. A screenshot of the interface for this interaction game is shown in Figure 6.3. Three players are instructed to perform a particular interactive behavior. The position of each player is recorded at 8 samples per second. Each behavior is recorded for approximately 30 seconds, $T \approxeq 240$.

The players are first instructed to follow each other in order. That is, player 1 is told to move their marker around the screen while player 2 is instructed to follow player 1. Player 3 is instructed to follow player 2. Using a uniform prior on structure and weak matrix - normal - inverse - Wishart prior on parameters, the posterior on structure is calculated given this data using a TIM(1). For increasingly restrictive sets of allowable structures results are shown in Figure 6.4(a).

Visualization of the uncertainty over interaction structure is challenging in general. This fact will motivate the use of more focused probabilistic evaluations as the data sets we consider grow in complexity. Here, we visualize the resulting posterior as weighted interaction graphs. In these graphs, edge intensity represents the posterior probability of that edge (darker is higher). Node intensity represents the probability a time-series has no parents and is a root. White indicates a probability of 0 while black indicates

a probability of 1. Note that this type of following behavior is described well by all sets of structures we consider. The posterior is the most peaked (exhibits the least uncertainty) when considering the most restrictive set, directed trees $\mathcal{T}_3$.

In a different sequence players 2 and 3 are instructed to move freely, while player 1 is told to follow player 2. The resulting posteriors are shown in Figure 6.4(b). While the posterior on structure agree with one's intuition, there is some uncertainty as to wether or not the edge $1 \rightarrow 2$ exists. Given the short sequence of data, this uncertainty may be related to disambiguating "1 following 2" versus "2 evading 1". Note this behavior is not described well by a directed spanning tree since player 3 is independent of all others.

Lastly, players 2 and 3 are told to move freely while player 1 does his best to stay between both of them. The results in Figure 6.4(c) show that this behavior only seems represented well in set $\mathcal{A}_3$; it is the only set that allows more than a single parent for a time-series. Restricting the the structures to have no more than one parent leads to the posterior using sets $\mathcal{P}_3^1$ and $\mathcal{F}_3$ to strongly favor an independent explanation. This independence is not allowed in the set of trees, $\mathcal{T}_3$, and thus the posterior exhibits the most uncertainty.

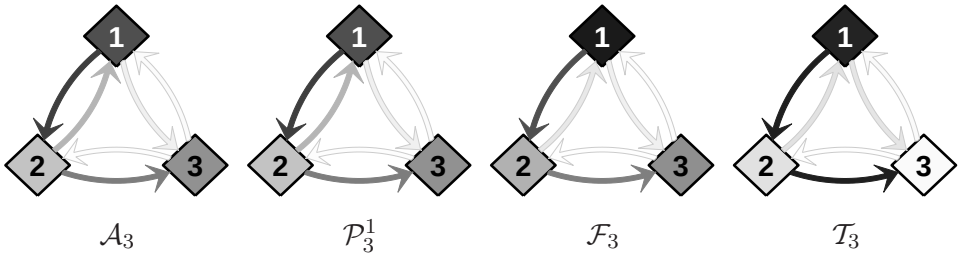
■ 6.2.2 Dynamic Interactions

Next we consider situations in which interaction changes over time. The first set of experiments, again, involve data collected from our interaction game. However, this time the players are told to switch among a specific set of behaviors. The second set of experiments explores data obtained from a recorded basketball game.

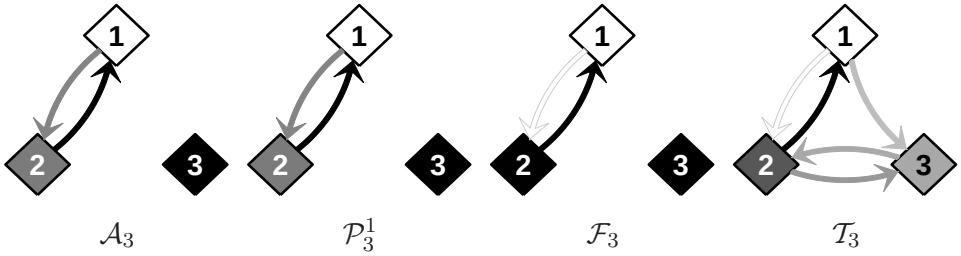
Follow the Leader

Using the same interactive game setup, we record three individuals playing a game of follow the leader. One player is designated the leader. The leader moves his or her marker randomly around the screen while the other players are instructed to follow. Here, the designated leader changes throughout the game. That is, a fourth person observing the game tells the players when to switch leaders. In this case, the latent variable indicating the change of leader is known, and consequently, nominal ground truth is available by which to evaluate performance.

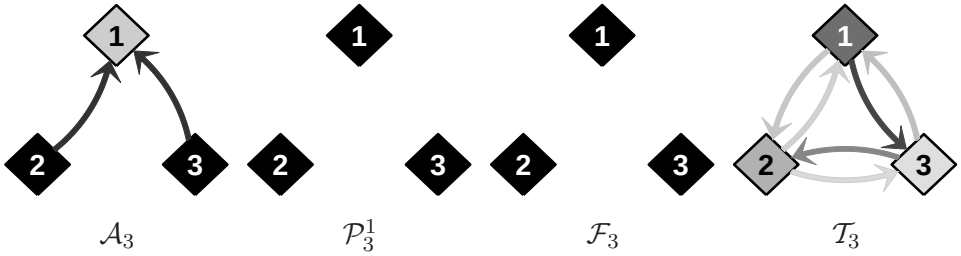
We begin by performing dynamic dependence analysis on this sequence using a switching temporal interaction model STIM(1,3) with $\bar{E} \in \mathcal{T}_3$. That is, we will use



(a) 3 follows 2 follows 1



(b) 1 follows 2



(c) 1 stays between 2 and 3

Figure 6.4. *Resulting Posterior Interaction Graphs for Specific Behaviors:* The results for each specific behavior enacted by three players using the "Interaction Game" are shown as weighted interaction graphs. The darkness of a node in the graph is probability of that time-series being a root. The darkness of an edge is a function of the probability of that edge being present. Black represents probability 1 while white is probability 0. Each subfigure/row is a separate behavior. Each column shows results when $\bar{E}$ is restricted to be a specific set. From left to right the columns represent increasingly restricted sets starting with the set of all structures to the set of directed trees.

the fact there are only three possible leaders and knowledge that directed trees may sufficiently describe the interaction among players. A uniform prior on structures and equivalently weak prior on parameters is used. A weak self biased prior on the state transition distribution is imposed with a bias towards self transition.

Given the data and the prior model, 100 samples of the structure, parameters and the hidden state sequence are obtained with a Gibbs sampler as described in Section

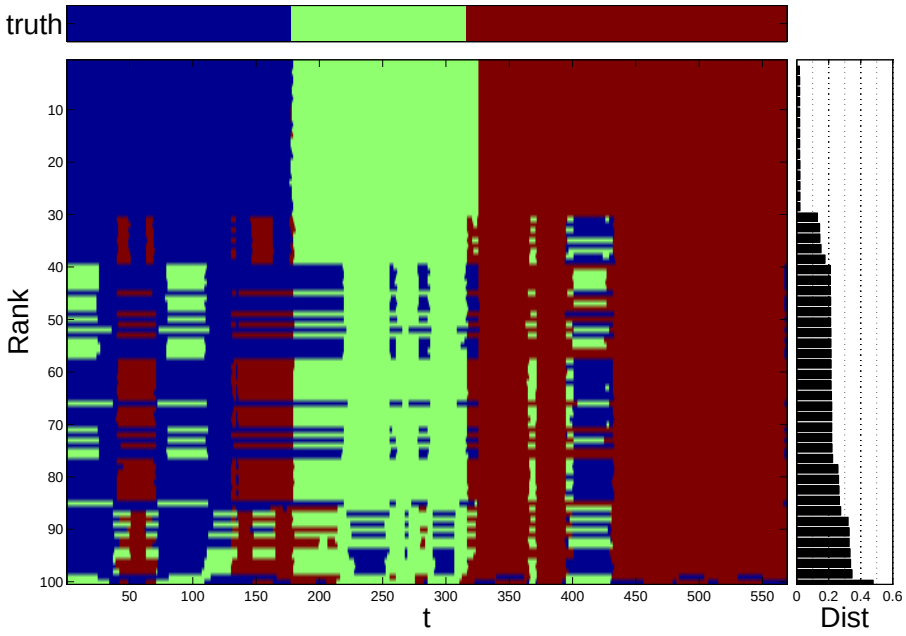


Figure 6.5. *Samples of z Using a STIM(1,3) on the Follow the Leader Dataset:* The very top plot shows the ground truth with color indicating who was designated the leader. Leaders changed in order from player 1 (blue) to player 3 (red). 100 state sequences drawn from our Gibbs sampler are displayed below the ground truth. The state sequence samples are ranked according to their log likelihood with the top state sequence representing the sample with the highest likelihood. The state sequence labels were permuted such that they minimized the hamming distance to the ground truth. This provides a consistent coloring. The hamming distance for each state sequence sample is shown on the right. The hamming distance is highly correlated with the log likelihood.

4.4. The two primary steps of the sampler are:

1. Jointly sampling a state sequence condition on the data and the current sample of parameters and structure.

2. Jointly sample the structure and parameters using exact updates of the posterior conditioned on the data and the sampled state sequence.

Burn-in required approximately 60 iterations, with little change in the sampled state sequence or data likelihood in future iterations. A detailed view of the results can be found in Figure 6.5. The ground truth state sequence is shown on top with players

taking turns being the leader in order. This figure shows each of the Gibbs sampled state sequences. The labels were permuted to give a consistent coloring with the ground truth segmentation shown on the top. The results are ranked by the log likelihood of the data given the sampled parameters and structures, with the top being the most likely. The side of the plot shows the normalized Hamming distance of the best mapping to the ground truth [43]. Note that this error metric is highly correlated with the log probability of the data. Each sample falls in one of two general categories. The top third of the samples match the ground truth closely, the bottom two thirds suggests a consistent alternative explanation.

By themselves, the state labels only indicate a change of distribution over structure. While this segmentation information is useful and interesting, we are interested in the details of the distribution. Given this segmentation we look at the posterior distribution on structure to analyze the interaction among the players. Figures 6.6 and 6.7 show a more detailed breakdown of two sampled models. The first row of each figure shows the most likely segmentation given the model. The second row shows weighted interaction graphs representing the posterior probability of the structure for each state. Recall that while the state sequence is sampled using an MCMC method, via the details outlined in Section 3.4.3, we can obtain an exact posterior conditioned on this sequence.

Figure 6.6 is a sample with low Hamming distance and high log likelihood. Notice that the posterior distribution on structure for each state is essentially a delta function on three distinct structures. These structures agree with our intuition in that each root is consistent with who was designated as the leader and the followers are conditionally independent given the root.

Figure 6.7 is a sample with a mid-range Hamming distance (ranked 34 out the 100 samples). It has errors consistent with the majority of sequences shown in 6.5. The confusion between the first and third state is most noticeably reflected in the structure posterior for state 3. The above analysis assumed three states, consistent with our knowledge of the ground truth, Figure 6.7 gives evidence for additional states. That is, for each phase of the game a better model may be a mixture of processes each with similar structure but different parameter distributions. In order to test this hypothesis, we repeat the experiment using $K = 6$ states. Figure 6.8 is a sample from this model.

The second row shows the occurrence of the learned states. We see that the ambiguity in Figure 6.7 is resolved by splitting the first and third state into two different models, each with a posterior on structure consistent with the ground truth. Interest-

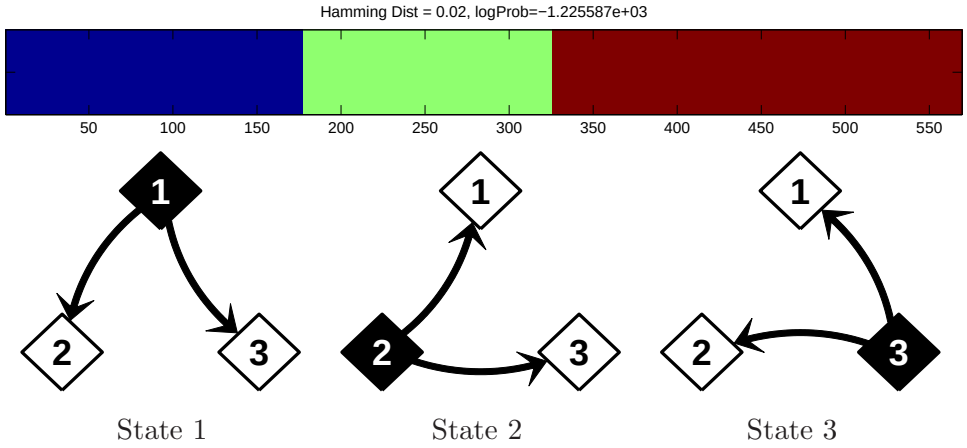


Figure 6.6. *Highest Log Likelihood Sample for Follow the Leader Data:* The sampled state sequence with the highest log likelihood out of 100 samples drawn using a STIM(1,3) is shown at the top. The tree graphs under the state sequence depict the posterior over structure for each state in terms of a weighted interaction graph. The states are lined up with the sampled state sequence. Note that this sample closely matches the ground truth and the posterior on structure for each state peaked at a structure which is indicative of who was leading.

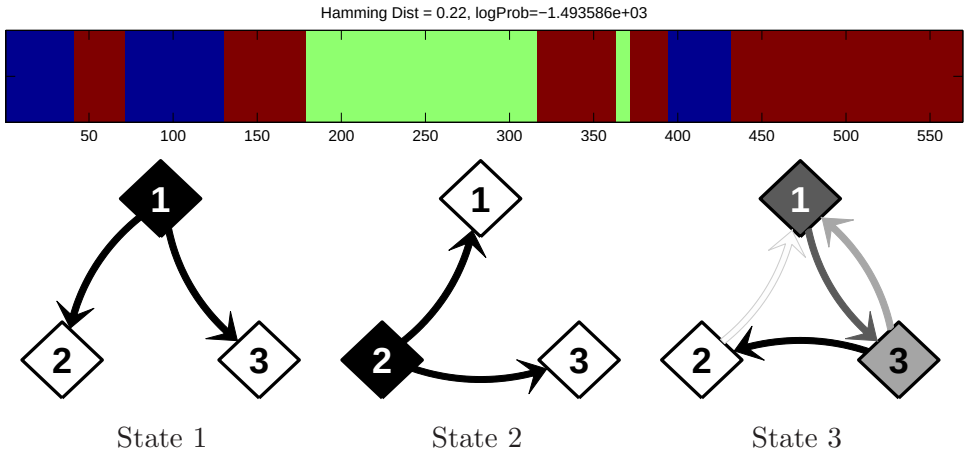


Figure 6.7. *Typical Sample for Follow The Leader Data:* Here the the sample state sequence ranked 34th in terms of log likelihood out of 100 samples drawn using a STIM(1,3) is shown at the top. The tree graphs under the state sequence depict the posterior over structure for each state in terms of a weighted interaction graph. The states are lined up with the sampled state sequence. Note there is some disagreement with the ground-truth and confusion between states 1 and 3. This reflected in some posterior uncertainty in the structure for state 3.

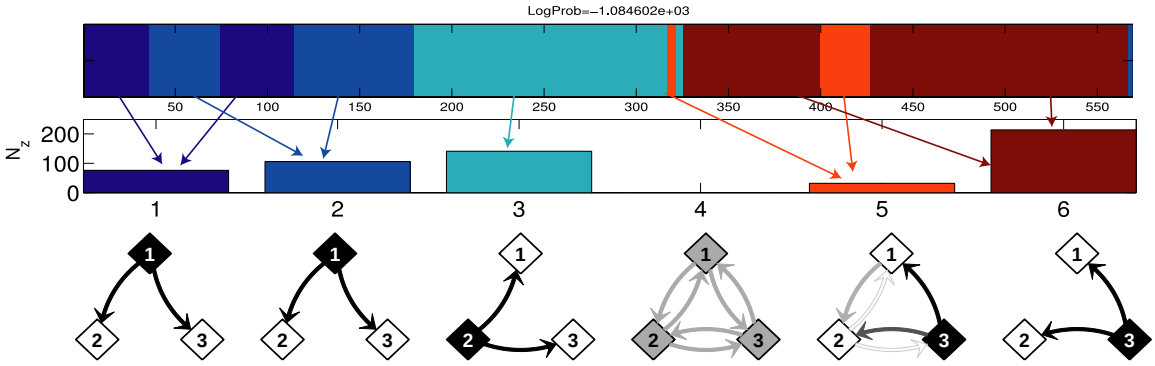


Figure 6.8. *Sample Drawn using a STIM(1,6) on the Follow the Leader Data:* The top row displays the sampled state sequence. The second row shows the amount of time each state was active in this sampled sequence. The third row shows posterior interaction graphs for each state aligned with the states in the middle row. Note that this sample explained player 1 being the leader with two states, each with a strong posterior time-series 1 being the root. States 5 and 6 both occur when player 3 is the leader with some uncertainty in structure for state 5. State 4 is never used in the sample and thus the posterior over structure is uniform for that state.

ingly, state 4 indicates uniform uncertainty in structure. However, this state is never used and our prior is uniform, thus its posterior distribution remains uniform. This suggests that perhaps 5 states were sufficient to capture the dynamic behavior of the observations.

Basketball

Next we consider analysis of player interactions in sports data. Automatic analysis of player interactions could be a valuable tool for understanding and learning individual player and team strategies. That is, information about the pattern and structure of interactions among teammates and their opposition could be used to learn a playbook for each team. Here, we focus on the core task of characterizing player interactions and leave the lofty goal of extracting strategic information for future work.

We explore a basketball game recording from the CVBASE 06 dataset [75]. Players are tracked in two cameras and their positions are mapped to a common coordinate system and recorded at each frame. Tracking is performed using a simple template correlation tracker which is corrected by hand using an interactive tool. A total of 11 tracks are obtained; five players on each team plus the ball. A coarse annotation of the

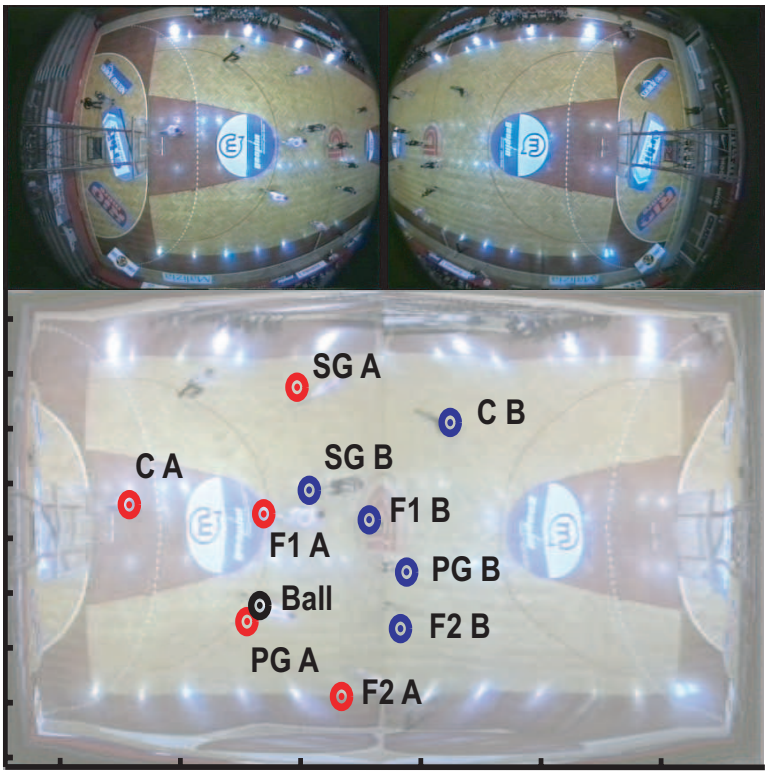


Figure 6.9. *Sample Frame from Basketball Data:* The top two images show the raw input from the two cameras recording the game. The bottom image shows the un-warped common coordinate frame with each player labeled.

current phase of the game is also created. Four phases are noted: team A on offense, team B on offense, team A transitioning to offense, and team B transitioning to offense. A sample frame with player positions is shown in Figure 6.9.

A STIM(2,10) model with $\bar{E} \in \mathcal{T}$ is used for analysis; a second order model incorporates velocity information. Again, we use a weak prior on parameters and uniform prior on structures. Note that here, with 11 time-series and $\bar{E} \in \mathcal{T}$, we are reasoning over 11^{10} structures for each state. However, as shown in Chapter 3, given a state, one can reason over these structures with approximately 11^3 operations. That is, we can reason over 25 billion structures with, on the order of, 2 thousand computational steps.

A sampled state sequence obtained from a Gibbs sampler is shown in the bottom row of Figure 6.10. The middle row shows the best many-to-one mapping of the sampled state sequence to the coarse annotation. That is, it uses a mapping which minimizes

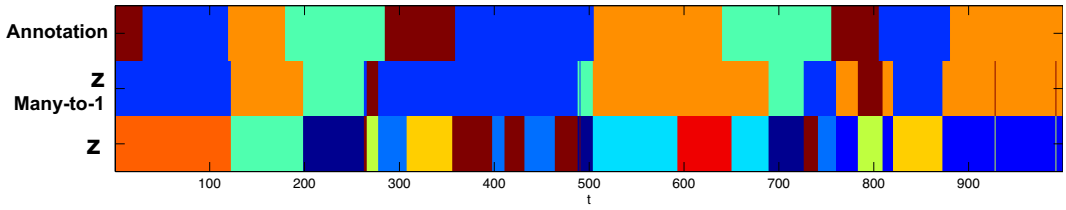


Figure 6.10. *Sampled State Sequence for the Basketball Data:* Sampled z from STIM(2,10) compared with the annotation describing which phase of the game is currently active. The top row shows the annotation which each annotated state using a different color. The bottom row shows the sampled state sequence z. The middle row shows the many-to-one mapping from the sampled sequence to the annotation that minimizes hamming distance.

Table 6.1. *Basketball Results Showing Probability of Root:* The probability of being a root during the four annotated phases of the game is shown for each team and the ball. **bold** and underlined values are the maximum and second highest in each column respectively

	A on offense	B on offense	B transitioning to offense	A transitioning to offense
Team A	.2475	.0832	.0420	.4769
Team B	.0468	.2902	.4234	.0459
Ball	**.7057**	**.6266**	**.5346**	**.4771**

the hamming distance. Note that while the sampled sequence is somewhat predictive of the annotation, a direct comparison is misleading as within each phase of the game multiple structures may be active.

Given a sampled state sequence, the posterior on $\bar{E}$ is obtained for each point in time. With 11 time-series and 10 states, displaying all posterior interaction graphs is impractical. As an alternative, we focus on calculating posterior event probabilities over time intervals. Table 6.1 shows the probability of the root being on either team or the ball given each phase in the coarse annotation. Over all phases of the game the ball has the highest probability of being the root. When on offense or transitioning to offense a player on team A has a higher probability of being the root than one on B and vice versa. Similarly, Table 6.2 shows the probability of being a leaf averaged over each team and the ball. Here the ordering is reversed and has a connection to which team is on defense. We see how analysis of these posterior event probabilities can give one a statistical prediction of the state of the game.

Table 6.2. *Basketball Results Showing the Average Probability of Being a Leaf:* The average probability someone on Team A, Team B, or the ball is a leaf during the four annotated phases of the game is shown. **bold** values are the maximum in each column respectively

	A on offense	B on offense	B transitioning to offense	A transitioning to offense
Team A	.3942	**.4097**	**.4880**	.3254
Team B	**.5515**	.0493	.4234	**.6512**
Ball	.1685	.0671	.1133	.2986

Lastly, we calculate the expected number of children for each player and the ball over all time. As discussed in Section 3.4.3, the number of children can be expressed as an additive function and the expectation is calculated using Equation 3.106. The additive function f used to calculate the number of children of time-series v is set such that $f_{\mathbf{S},u} = 1$ if $v \in \mathbf{S}$ and 0 otherwise. The top four time-series, in terms of expected number of children, are the ball, point guard A (PG A), forward 1 A (F1 A), and forward 1 B (F1B) with expectations of 1.76, 1.73, 1.08, and 0.95. Not surprisingly, the ball has the largest influence on the dynamics of the game. The point guard generally controls the flow of the game and is usually the best ball handler.

As a consequence of the framework developed in Chapters 3 and 4 we can pose more complex probabilistic questions with regard to the posterior distribution over interaction structures. For example, now that we know that point guard A has significant influence we can look at his interactions in more detail. Figure 6.11 shows the posterior probability of an edge from the PG A to every other player over all time given a sampled state sequence. Notice that the switching among states can be seen in terms of changes in edge posteriors. PG A tends to influence his own forward as well as the point guard and forward on the other team.

■ 6.3 Summary

In this chapter, we have cast the problem of object interaction analysis in terms of static and dynamic dependence analysis. Vector autoregressive TIM(r)s were used to described causal relationships among observed trajectories of objects and the priors defined in Chapter 3 allowed for tractable Bayesian inference.

We illustrated the utility of our framework via experiments characterizing posterior

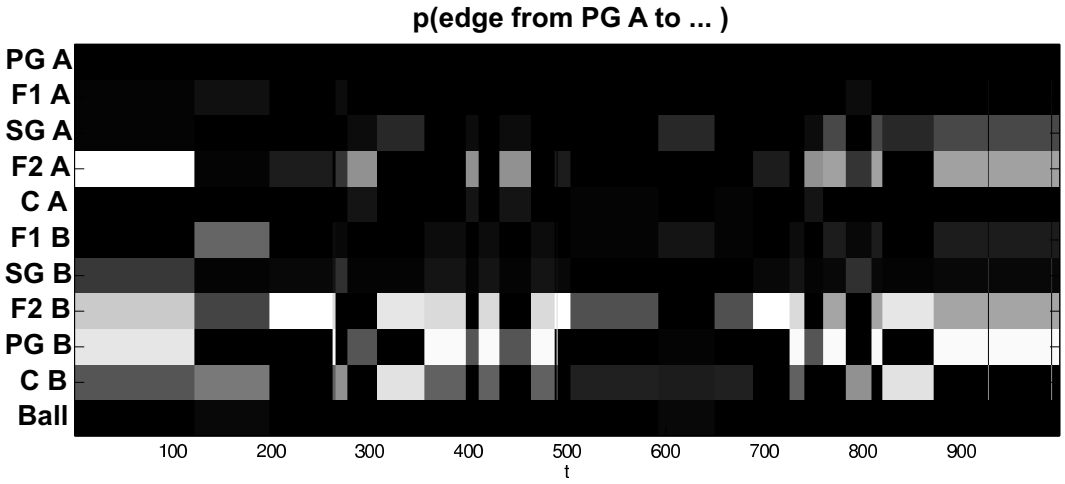


Figure 6.11. *Influence of Point Guard A (PG A):* Using a sampled state sequence each row represents the amount of influence PG A has on a specific player as a function of time. Influence is expected number of edges leaving PG A's time-series. Lighter colors mean higher probability of influence.

structural uncertainty among the observed object trajectories. Static dependence analysis was performed on the output of a simple multi-person interactive computer game and results were obtained which were consistent with one's intuition.

Using a STIM we were able to analyze data in which interactions changed over time. The posterior uncertainty in structure revealed the potential for improvements to our model by hypothesizing more interactive states. Lastly, we analyzed data from a real basketball game. While we are not at the level of discovering playbooks, detailed structural inference using our model yielded posterior statistics predictive of who was on offense and defense. In addition, further analysis identified the key influential players and how their influence changed over time.

Chapter 7

Conclusion

In the preceding chapters we have developed a framework for analyzing changing relationships among multiple time-series. We cast this dynamic dependence analysis task in terms of inferring the underlying structure of probabilistic models describing observed time-series. By doing so we were able to leverage a large body of existing work on statistical modeling and structure inference. Motivated by two distinct problems of audio-visual association and object interaction analysis, our primary focus was on structural inference and description of uncertainty in inferred dependence rather than learning and recognition. That is, our goal was to describe the dependence among observed time-series in the absence of training data rather than building predictive models for future recognition tasks.

7.1 Summary of Contributions

We introduced two static dependence models for time-series.

A FactM allows one to model the evolution of multiple time-series in terms of independent groups. A TIM allows one to model detailed causal relationships among multiple time-series.

We presented a decomposable conjugate prior on the structure and parameters of a TIM.

Taking advantage of temporal causality, this prior allows a super-exponential number of structures to be reasoned over in exponential-time in general. Restricting the class of structures such that each time-series has a bounded number of influences allows a still super-exponential number of structures to be reasoned over in polynomial-time. Conjugacy yields tractable calculation of exact joint and marginal posterior probability of structure in addition to posterior expectations.

Furthermore, an exact characterization of uncertainty can be tractably obtained with nuisance parameters integrated out.

We introduced a DDM for reasoning about dependence relationships changing over time.

A DDM extends the static dependence models presented in Chapter 3 to allow for changing dependence structure. Building on the large body of work on dynamic Bayesian networks and HMMs, one can reason over an exponential number of sequences of dependence relationships in linear-time. Inference using a DDM can exploit past and future data when making a local decision about dependence. We showed both theoretically and empirically how this property yields advantages over standard windowed analysis.

We demonstrated state-of-the-art performance on an audio-visual speaker association task.

This performance was achieved without the benefit of training data or a silence detector. In addition, there were no window size or threshold parameters to set. The semantic label of who is speaking was naturally mapped to a specific structure of association among the observed audio and video time-series using a FactM.

We demonstrated the utility of dynamic dependence analysis to characterize the interaction of multiple moving objects.

Object interaction analysis was formulated in terms of inference over TIM dependence structure. A conjugate prior allows one to fully characterize posterior structural uncertainty among observed object trajectories. Results obtained when analyzing simple multi-person interactive computer games were consistent with the behavior of the instructed participants. Analyzing the trajectories of basketball players revealed key influential players and who they influenced over time. Additionally, posterior statistics on structure were predictive of the overall state of the game.

■ 7.2 Suggestions for Future Research

In this dissertation we discussed the general concept of a dynamic dependence analysis task, presented two specific dependence models and explored their use in two distinct

applications domains that best matched their strengths. We believe our models and extensions thereof can be useful for a wide variety of other problems. Below we overview some possible extensions and open research directions for future work.

7.2.1 Alternative Approaches to Inference

In Chapter 3 we presented two approaches to inference. For the FactM we used a maximum likelihood approach, while for the TIM we presented priors which allowed for exact Bayesian inference. The choice of which approach to use was based primarily on the end applications of interest. In audio-visual speaker association task, our end goal was to make a decision about who was speaking. Thus, point estimates obtained from an ML approach using a FactM were suitable. However, one could also place a discrete prior on the set of factorizations of interest, along with a conjugate prior on parameters for the FactM, and calculate the MAP structure. Some initial experiments using MAP estimates of structure on the CUAVE dataset yielded comparable performance to the ML approach. A more extensive analysis and experiments on other datasets would be beneficial to those interested in the audio-visual association problem.

In the object interaction analysis task our primary goal was to explore a large set of structures and give a full characterization of dependence. Thus, using a Bayesian approach was well justified. However, an ML approach could be also applied to this problem. A combination of using exact Bayesian inference over structure with ML point estimates of parameters is another alternative that could be explored.

7.2.2 Online Dynamic Dependence Analysis

We defined dynamic dependence analysis as a batch analysis task. This allowed one to combine past and future information to make a local decision about dependence relationships. The end applications we explored were in this batch context: providing metadata to describe who was speaking in an archived video and post analysis of interactions among tracked objects.

A future avenue of research is to extend our approach to allow for online inference of dependence relationships. Since our DDM is simply an HMM at the highest level, one can take advantage of the large body of existing work on online learning and inference in dynamic Bayesian networks (c.f. [67, 27]). As briefly mentioned in Chapter 4, our static dependence models can also be embedded in other alternative dynamic models such as the Parts Partition model for which efficient online Bayesian inference algorithms have

been developed [26, 28]. It is important to note that an online approach will still have an advantage over windowed approaches in that it can incorporate past information when performing inference.

■ 7.2.3 Learning Representation and Data Association

As mentioned in our introduction a key challenge in fusing information from multiple sources is that of representation. In this dissertation we assumed a sufficiently informative representation was given to us and focused on identifying dependence among sources. An interesting path for future research is explore efficient ways to simultaneously learn the representation and structure of dependence. We expect that knowledge of which sources of data are dependent and when could help one learn more informative representations.

Similarly, we assumed each observed time-series was consistent and represented a single underlying source over all time. That is, we assumed that data association was solved. Specifically, we assumed that a tracker provided consistent moving object trajectories. Obtaining consistent trajectories is a difficult task for tracking systems, particularly when objects are close together and may occlude one another. Thus, one may wish to jointly track and reason about dependence. Knowing which objects are interacting (and how) should help a tracker better predict future positions and obtain more consistent tracks.

■ 7.2.4 Latent Group Dependence

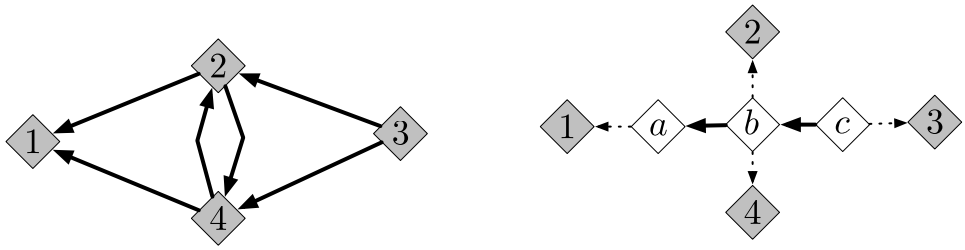


Figure 7.1. *Latent Group Dependence:* True interaction model for 4 time-series (left). Grouping representation (right) where there are 3 latent groups a,b, and c. Group b "clusters" streams 2 and 4 together. Solid arrows represent temporally causal influence from the past to the future. Dotted arrows represent an instantaneous relationship.

Our static dependence models allowed one to reason over the dependence among

a set of observed time-series. They did not consider the potential for unobserved factors causing dependence among what was observed. Consider a parade with multiple floats/vehicles following each other down a street. Each float has a group of many people walking next to it. If we were given trajectories for the people and not the floats, the inferred dependence relationships might be overly complex. That is, each person would appear to be influenced by the others surrounding the same float and at the same time be influenced by the people surrounding the float ahead. If the trajectories of the floats were also observed the less complex description may emerge. Each float can be described as being influenced by the float ahead and each person being conditionally independent given the float they are surrounding. Similarly, when analyzing the interaction of many people in a crowd , it may be advantageous to model interactions among groups of people rather than interactions among individuals.

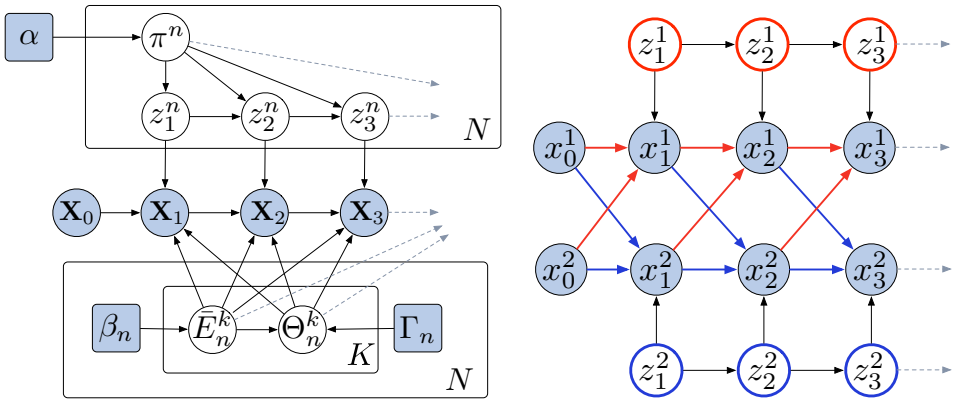
Figure 7.1 depicts a scenario in which one observes 4 time-series. If these time-series correspond to positions of moving objects, the left portion of the figure shows an interaction graph that may emerge when object 3 is being followed by objects 2 and 4 moving together as a group with object 1 following them. The right portion of the figure shows an alternative view in which three latent groups or clusters are considered.

A latent group dependence model can be designed in which each of the N observed time-series are assigned a label associated it with one of M groups. Each latent group is represented as a time-series. Each observed time-series is modeled to be conditionally independent given their group assignments. Given samples of the latent group time-series one can reason about their dependence using one of the static dependence models presented in this dissertation. Such a group dependence model would require a more complex inference procedure. In a Bayesian inference task a Dirichlet prior could be placed on group assignments and a Gibbs sampler used to generate samples from the posterior. This latent group dependence model could be further extended to allow group assignments, parameters and structure to change over time.

■ 7.2.5 Locally Changing Structures

The latent states used in our DDM index into a finite set of structures and parameters. A change in latent state value over time indicated a global change in structure. If one removed global constraints on dependence structure, an alternative model could be designed in which each time-series has its own latent state z_t^v. The latent state for time-series v could specify which, if any, of the other time-series it depends on.

If there are N time-series, each with its own latent state taking on K values such a model could reason over K^{N^T} possible sequences of dependence structures. While the DDM presented in this dissertation with K^N states could also model as many sequences, a locally switching model has the advantage of being able to pool information from times which share the same local structure. Figure 7.2(a) shows a general graphical model for allowing local dependence changes in a TIM. A specific example with $N = 2$ is depicted in Figure 7.2(b) in which each latent state is a binary value indicating whether a particular edge is present or not. Color is used to indicated which latent states are controlling which edges.



(a) (b)

Figure 7.2. *Graph Models for the Proposed Local Dependence Switching:* (a) The high level structure of the model. (b) A specific example with $N = 2$ and $K = 2$ two possible states. z_t^1 controls the presence of (red) edges going into x_t^1 while z_t^2 controls (blue) edges going into x_t^2.

■ 7.2.6 Long Range Temporal Dependence

The static dependence models presented in this dissertation assumed $r-th$ order temporal dependence. The current value of each time-series could only be directly dependent on the value other time-series within the window of r past time points. This restriction was reasonable for the applications we looked at. However, consider using a TIM to model a teacher student relationship among moving objects. That is, imagine one teacher object demonstrating a behavior for a student object. The student can observe the teacher and then recreate the behavior. The student object is clearly dependent on the teacher, but this dependence is delayed by some lag l. In order to capture this

relationship using a TIM, the temporal order r must be set to be greater than or equal to l. If l is large, this will result in an overly complex model since a TIM assumes dependence on all r past values. Thus, a useful modification of the TIM to explore in the future may be one that can incorporate lags such that:

$$p\left(\mathbf{X}_t|\widetilde{\mathbf{X}}_t, L, \bar{E}, \Theta\right) = \prod_{v=1}^{N} p\left(\mathbf{x}_t^v|\widetilde{\mathbf{x}}_t^v, \widetilde{\mathbf{x}}_{t-l^v}^{\mathbf{pa}(v)}, \Theta_{v|\mathbf{pa}(v)}\right), \tag{7.1}$$

where $L = \left\{l^1, \ldots, l^N\right\}$ is a set of lags for each time-series. Each time-series is dependent on its own r-th order past $\widetilde{\mathbf{x}}_t^v$ and the r-th order past of a parent set of time-series lagged by l^v. A distribution can be placed over these lags $p(L)$. Depending on the form of the lag distribution inference may still be tractable in a Bayesian setting. That is, if one only considers a small set of possible discrete lags, the complexity of inference will only increase linearly with the size of this set.

■ 7.2.7 Final Thoughts

Modeling and understanding evolving dependence structure has been a challenging and intriguing research topic. While this dissertation has focused on two specific applications, we believe the ideas and models presented here can be readily adapted and extended to a wide variety of other domains. Stepping back and reassessing ones own research is often a humbling experience. It is easy to question the meaning or impact of what many times turns into a very personal endeavor. I only hope the models and ideas presented in this dissertation will be helpful to others in their own research pursuits.

Appendix A

Directed Structures

In Section 3.4.2 we discussed a prior on the directed structure $\bar{E}$ which encodes the dependence relationships in a temporal interaction model (TIM). The prior has the following form:

$$p_0\left(\bar{E}\right) = \frac{1}{Z(\beta)} \prod_{v=1}^{N} \beta_{\mathbf{pa}(v),v}. \tag{A.1}$$

In Section 3.4.3 we showed that this prior is conjugate and thus the posterior takes the same form. In this Appendix we detail methods for sampling $\bar{E}$ using this prior and/or the posterior. In addition, we discuss how to sample parameters for a TIM given a structure $\bar{E}$, how to obtain the MAP structure, and discuss potential numerical issues.

A.1 Sampling Structure

We begin by outlining how to sample $\bar{E}$ given it has the distribution shown in Equation A.1. First, we discuss the sampling procedure for when $\bar{E}$ has no global constraints (i.e. $\bar{E} \in \mathcal{A}_N$ or $\bar{E} \in \mathcal{P}_N^K$). Next, we discuss how to sample $\bar{E}$ when it is restricted to be a directed tree ($\bar{E} \in \mathcal{T}_N$) or forest ($\bar{E} \in \mathcal{F}_N$).

A.1.1 Sampling Without Global Constraints

If there are no global constraints on edges in the structure $\bar{E}$, the parent set for each of the N time-series can be sampled independently. By enumerating all valid parent sets for each time-series one can sample a parent set using a discrete distribution. That is, let $\mathbf{R}_v^{(i)}$ be the i-th possible parent set for time-series v. The index i can range from 0 to the total number of possible parent sets, q_v. A parent set can be then be sampled using a discrete distribution over q_v possible values. For example, if $N = 3$ and $\bar{E} \in \mathcal{A}_N$

(can be any directed structure), then $q_1 = q_2 = q_3 = 4$ and

$$\mathbf{R}_1^{(1)} = \{\}, \quad \mathbf{R}_1^{(2)} = \{2\}, \quad \mathbf{R}_1^{(3)} = \{3\}, \quad \mathbf{R}_1^{(4)} = \{2,3\} \tag{A.2}$$

$$\mathbf{R}_2^{(1)} = \{\}, \quad \mathbf{R}_2^{(2)} = \{1\}, \quad \mathbf{R}_2^{(3)} = \{3\}, \quad \mathbf{R}_2^{(4)} = \{1,3\} \tag{A.3}$$

$$\mathbf{R}_3^{(1)} = \{\}, \quad \mathbf{R}_3^{(2)} = \{1\}, \quad \mathbf{R}_3^{(3)} = \{2\}, \quad \mathbf{R}_3^{(4)} = \{1,2\}. \tag{A.4}$$

If $N = 3$ and $\bar{E} \in \mathcal{P}_N^1$ (has at most 1 parent), then $q_1 = q_2 = q_3 = 3$ and

$$\mathbf{R}_1^{(1)} = \{\}, \quad \mathbf{R}_1^{(2)} = \{2\}, \quad \mathbf{R}_1^{(3)} = \{3\} \tag{A.5}$$

$$\mathbf{R}_2^{(1)} = \{\}, \quad \mathbf{R}_2^{(2)} = \{1\}, \quad \mathbf{R}_2^{(3)} = \{3\} \tag{A.6}$$

$$\mathbf{R}_3^{(1)} = \{\}, \quad \mathbf{R}_3^{(2)} = \{1\}, \quad \mathbf{R}_3^{(3)} = \{2\}. \tag{A.7}$$

Algorithm A.1.1 outlines the sampling procedure. Note that the sampling procedure requires one to enumerate all allowable parent sets for each $v \in \{1, \ldots, N\}$. When all structures are allowed, $\bar{E} \in \mathcal{A}_N$, each v has 2^{N-1} allowable parent sets that must be indexed into. In practice, we store a bidirectional mapping between an N bit number representing a parent set and an index. The non-zero bits in the N bit number indicate which time-series are parents. When $\bar{E} \in \mathcal{A}_N$ this map is straightforward and one can use the N bit number itself as the index. When the size of the parent set is restricted, $\bar{E} \in \mathcal{P}_N^K$, we form the map explicitly by recursively looping over all parent sets of size 0 to K.

■ A.1.2 Sampling Directed Trees and Forests

When $\bar{E}$ is restricted to be a directed tree or forest one can no longer treat each parent set independently. Fortunately, sampling directed trees is a well studied problem and one can easily transform existing procedures for trees to also sample directed forests. Most directed tree sampling algorithms can be categorized as either determinant or random walk based approaches. All approaches can be initialized by randomly choosing a root based on Equation 3.97.

Determinant based approaches such as [19] randomly choose an edge based on the edge probabilities calculated using the Matrix Tree Theorem, subsequently contracting chosen edges until a tree is formed. A straightforward implementation has a complexity of $O(N^4)$ while improvements can be made to reduce the running time to $O(M(N))$ where $M(N)$ is the complexity of multiplying two $N \times N$ matrices.

Random walk based algorithms simulate a random walk on a stochastic graph defined by β starting at a randomly sampled root. While walking on the graph one notes

Algorithm A.1.1 Sampling $\bar{E}$ Without Global Constraints

Require: The full set of parameters β, and an enumerated set of all possible parents for each of the N vertex (time-series) $\mathbf{R}$.

function SAMPLEEBARWITHOUTGLOBALCONSTRAINTS($\beta, \mathbf{R}$)

% Start with no edges

$\bar{E} \leftarrow \{\}$

for $v = 1$ to N **do**

% Define probabilities for each possible parent set

$z \leftarrow \sum_{i=1}^{q_v} \beta_{\mathbf{R}_v^{(i)},v}$

for $i = 1$ to q_v **do**

$\pi_i \leftarrow \frac{1}{z}\beta_{\mathbf{R}_v^{(i)},v}$

end for

% Sample a parent set index

$k \sim \text{Discrete}\,(.; \pi_1, \ldots, \pi_{q_v})$

% Add the edges for the indexed parent set to v

for each $u \in \mathbf{R}_v^{(k)}$ **do**

Add edge (u, v) to $\bar{E}$

end for

end for

return $\bar{E}$

end function

each vertex that has been visited, erasing cycles as they are created until the walk has generated a tree. The algorithm presented by [93] has an expected running time proportional to the mean hitting time on the stochastic graph. The mean hitting time is the expected time to go from any random vertex to any other given the steady state distribution of the stochastic graph.

Both approaches have advantages and disadvantages. Wilson's algorithm is very simple to implement and for many graphs the mean hitting time is much less than N^3. However, it possible to have a β that yields graphs with exponentially large hitting times. While determinant based algorithms have fixed complexity, they are more

complex and require careful implementations to ensuring adequate precision [19]. We discuss related numerical issues further in Section A.4.

In this dissertation, we use a simple modification of Wilson's RandomTreeWithRoot procedure for sampling both directed trees and forests. Algorithm A.1.2 outlines the sampling procedure given β in matrix form. Note that in order to be consistent with Wilson's original description in [93] Algorithm A.1.2 samples a directed tree with all its edges reversed (*i.e.* the root is defined to have no children and leaves have no parents). We convert between this edge reversed tree and our desired form with the supporting functions shown in Algorithm A.1.3.

We modify Wilson's algorithm by setting a limit to the number of iterations.

This allows us to restart, try a different approach, or adjust our β to approximate our distribution with one which has a lower mean hitting time.

If we wish to sample a directed forest we can simply create a virtual root vertex $N+1$ and modify β such that the weights from node $N+1$ to any v is the root weight for v. That is set $\beta_{N+1,v} = \beta_{v,v}$ for all $v \in \{1, \ldots, N\}$. We can then call RandomTreeWithRoot(N+1,β,maxIter) to get a random forest.

■ A.2 Sampling Parameters

Given a sampled structure we can also sample parameters from the prior $p_0\left(\Theta|\bar{E}\right)$ or posterior $p\left(\Theta|\bar{E},\mathcal{D}\right)$. As discussed in Section 3.4.2 our prior on parameters is conjugate and both tasks involve sampling from the form:

$$\prod_{v=1}^{N} p_0\left(\Theta_{v|\mathbf{pa}(v)}|\Upsilon\right). \tag{A.8}$$

where the hyperparameters Υ are updated as a function of the observations $\mathcal{D}$ when obtaining the posterior. This modular and independent structure of the distribution allows one to sample each parameter independently as shown in Algorithm A.2.1.

Algorithm A.1.2 The RandomTreeWithRoot Algorithm with Iteration Limit

Require: β in matrix form with element i, j being $\beta_{i,j}$.

```
function RandomTreeWithRoot(r, β, maxIter)
    Ḡ ← MakeGBar(β)
    for i = 1 to N do
        InTree[i] ← false
    end for
    Next[r] ← nil
    InTree[r] ← true
    for i = 1 to N do
        Iter ← 0
        u ← i
        while not InTree[u] do
            Next[u] ← RandomSuccessor(u, Ḡ)
            u ← Next[u]
            Iter ← Iter +1
            if Iter > maxIter then
                failed
            end if
        end while
        u ← i
        while not InTree[u] do
            InTree[u] ← true
            u ← Next[u]
        end while
    end for
    Ē ← NextToEbar(Next)
end function
```

Algorithm A.1.3 The RandomTreeWithRoot Supporting Functions

function MAKEGBAR(β)
  % Transpose beta to reverse edge directions
  % This allows us to be consistent with Wilson's algorithm
  $\beta \leftarrow \beta^{\mathsf{T}}$
  for $v = 1$ to N **do**
    $\beta_{v,v} \leftarrow 0$
    $z \leftarrow \sum_{i=1}^{N} \beta_{v,i}$
    for $u = 1$ to N **do**
      $\bar{G}_{v,i} = \frac{1}{z}\beta_{v,i}$
    end for
  end for
  return $\bar{G}$
end function

function RANDOMSUCCESSOR(u,$\bar{G}$)
  $n \sim \text{Discrete}\left(.; \bar{G}_{u,1}, \ldots, \bar{G}_{u,N}\right)$
  return n
end function

function NEXTTOEBAR(Next)
  $\bar{E} \leftarrow \{\}$
  for $v = 1$ to N **do**
    if Next[v] not **nil then**
      % Note again, we are reversing the roll of Next
      Add edge (Next[v],v) to $\bar{E}$
    end if
  end for
end function

Algorithm A.2.1 Procedure for Sampling Parameters Given Structure

Require: A sampled structure $\bar{E}$ and hyperparameters Υ
for $v = 1$ to N **do**
  $\Theta_{v|\mathbf{pa}(v)} \sim p_0\left(\Theta_{v|\mathbf{pa}(v)}|\Upsilon\right)$
end for

A.3 Obtaining the MAP Structure

In some situations one may wish to obtain the MAP structure. That is, find

$$\bar{E}^* = \arg\max_{\bar{E}} \prod_{v=1}^{N} \beta_{\mathbf{pa}(v,\bar{E}),v} \tag{A.9}$$

$$= \arg\max_{\bar{E}} \sum_{v=1}^{N} \log \beta_{\mathbf{pa}(v,\bar{E}),v} \tag{A.10}$$

When no global constraints are imposed on $\bar{E}$ the MAP structure can be found by identifying the MAP parent set independently for each time-series. Thus, it the procedure's complexity is proportional to the N times the size the set of allowable parents for each vertex. Algorithm A.3.1 outlines this procedure.

Algorithm A.3.1 Obtaining MAP $\bar{E}$ without Global Constraints

Require: The full set of parameters β, and an enumerated set of all possible parents for each of the N vertex (time-series) $\mathbf{R}$.

function MAPEBARWITHOUTGLOBALCONSTRAINTS($\beta, \mathbf{R}$)

$\bar{E}^* \leftarrow \{\}$

for $v = 1$ to N **do**

$m \leftarrow$ -Inf

$z \leftarrow \sum_{i=1}^{q_v} \beta_{\mathbf{R}_v^{(i)},v}$

for $i = 1$ to q_v **do**

$\pi_i \leftarrow \frac{1}{z}\beta_{\mathbf{R}_v^{(i)},v}$

if $\pi_i > m$ **then**

$m \leftarrow \pi_i$

$k \leftarrow i$

end if

end for

for each $u \in \mathbf{R}_v^{(k)}$ **do**

Add edge (u, v) to $\bar{E}^*$

end for

end for

return $\bar{E}^*$

end function

■ A.3.1 Directed Trees and Forest

Finding the MAP directed tree or directed forest is well studied problem. Finding the MAP structure can be thought of as a maximum weight branching (forest) problem, a minimum weight arborescence (directed tree) problem, or a minimum weight rooted arborescence problem (c.f. [57]). Each of these problems are equivalent and algorithmic solutions were found independently by Chu and Liu [18], Edmonds [25] and Bock [10].

Here, for completeness, we provide the algorithm in the context of the maximum weight rooted directed tree problem. Given a set of directed edges $\bar{E}$, a designated root r and a positive weighting function $w(u,v) \in \mathbb{R}_+$ for each edge $(u,v) \in \bar{E}$ we wish to find the subset $\bar{E}^*$ which maximizes the sum of weights $\sum_{(u,v)\in\bar{E}^*} w(u,v)$. The steps of the solution are outlined Algorithm A.3.2. We assume there exists at least one directed tree from root r.

At a high level there are four phases. First, one greedily selects edges. If the result is a directed tree, the algorithm returns. If not, second, it finds a cycle in the resulting edge set and modifies the problem such that the cycle is represented by a new vertex and new edges and weights are added to and from this new cycle vertex. This step is outlined in Algorithm A.3.3. The third phase takes this modified graph and recursively calls the algorithm. The fourth phase takes the result from this recursive call and cleans up by expanding the contracted cycle node and remapping edges using book keeping established in Algorithm A.3.3[1]. A straightforward implementation has worst case performance $O(N^3)$. Each greedy edge selection step is $O(N^2)$ and one can recurse a maximum of N times. More efficient implementations such as [86] are $O(N^2)$.

We can map the problem of calculating the MAP directed tree ($\bar{E} \in \mathcal{T}_N$) to the maximum weight rooted directed tree problem. First, we use $logp_0\left(\bar{E}\right)$ and map $w(u,v) = \log \beta_{u,v}$. Note that while the log is likely to produce negative weights one can always shift the weights by a constant so they are non-zero as the solution is invariant to constant shifts (or equivalently multiplying the posterior by a constant). Second, we have to address the fact that we do not know the root. We can deal with this in two ways. One way is to simply call ChowLiuEdmondsBock for all N roots r and compare the results, pick the resulting directed tree with the highest probability.

Another approach is to add a virtual root node v_r and create an edge from v_r to all

[1] We found the algorithmic description and example in [61] to be helpful when implementing these functions. We only add them here for completeness and to be explicit about the extra book keeping steps.

Algorithm A.3.2 The Chu-Liu, Edmonds, Bock Algorithm

Require: A set of allowable edges $\bar{E}$ on vertices V, a designated root r and a weighting function w.

Ensure: The returned edge set $\bar{E}^*$ forms a directed spanning tree at root r with maximum weight $\sum_{(u,v)\in\bar{E}^*} w(u,v)$

function ChuLiuEdmondsBock$(V, \bar{E}, r, w)$

% **Greedy Seletion**

$\bar{E}^g \leftarrow \{\}$

for each $v \neq r$ in V **do**

$\bar{E}^g \leftarrow \bar{E}^g \cup \arg\max_{(p,v)\in\bar{E}} w(p,v)$ % **Add best incoming edge**

end for

if $\bar{E}^g$ has no cycles/loops **then**

return $\bar{E}^g$

end if

% **Cycle contraction and graph re-weighting**

Find a cycle C in $\bar{E}^g$

$\left\{V_c, \bar{E}_c, w_c, \Psi, \Phi, v_c\right\} \leftarrow$ Contract$(C, V, \bar{E}, w)$ % **A.3.3**

% **Recursive call on modified graph**

$\bar{E}^* \leftarrow$ ChuLiuEdmondsBock$(V_c, \bar{E}_c, r, w_c)$

% **Clean up edges into** C

$y \leftarrow \mathbf{pa}\left(v_c, \bar{E}^*\right)$, $z \leftarrow \Phi(y, v_c)$, $x \leftarrow \mathbf{pa}\left(z, C\right)$

$\bar{E}^* \leftarrow \bar{E}^* \setminus (y, v_c)$

$\bar{E}^* \leftarrow \bar{E}^* \cup (y, z)$ % **Add true incoming edge**

$\bar{E}^* \leftarrow \bar{E}^* \cup C$ % **Add back the cycle**

$\bar{E}^* \leftarrow \bar{E}^* \setminus (x, z)$ % **Break the cycle**

% **Clean up edges out of** C

for each y s.t. $(v_c, y) \in \bar{E}^*$ **do**

$z \leftarrow \Psi(v_c, y)$

$\bar{E}^* \leftarrow \bar{E}^* \setminus (v_c, y)$

$\bar{E}^* \leftarrow \bar{E}^* \cup (z, y)$

end for

return $\bar{E}^*$

end function

Algorithm A.3.3 Contract Function used by Algorithm A.3.2

Require: A cycle C, a set of vertices V, allowable edges $\bar{E}$ and a weighting function w

Ensure: Return a modified graph $\{V_c, \bar{E}_c\}$ in which all vertex in C have been collapsed into a single vectex v_c and edge weights are modified to be w_c. The edge weights are calculated such that finding the maximum spanning tree in this modified graph can be mapped back to the original problem using book keeping information Ψ and Φ.

function CONTRACT($C, V, \bar{E}, w$)

Let v_c be a new vertex representing C

$V_c \leftarrow V \setminus \text{vertices}(C) \cup v_c$

$\bar{E}_c \leftarrow \bar{E} \setminus C$

$w_c \leftarrow w$

% Deal with edge into C

$T_c \leftarrow \sum_{(u,v)\in C} w(u,v)$ **% Total cost of cycle**

for each vertex $y \notin C$ s.t. there exists an edge (y,z) with $z \in C$ **do**

$z \leftarrow \arg\max_{z'} w(y,z') + T - w(\mathbf{pa}(z', C), z')$

$w_c(y, v_c) \leftarrow w(y,z) + T - w(x,z)$

$\bar{E}_c \leftarrow \bar{E}_c \cup (y, v_c)$

$\Phi(y, v_c) \leftarrow z$ **% Remember z is the true end of this edge**

end for

% Deal with edges out of C

for each vertex $y \notin C$ s.t. there exists an edge (z,y) with $z \in C$ **do**

$z \leftarrow \arg\max_{z'} w(z', y)$

$w_c(v_c, y) \leftarrow w(z,y)$

$\bar{E}_c \leftarrow \bar{E}_c \cup (v_c, y)$

$\Psi(v_c, y) \leftarrow z$ **% Remember z is the true start of this edge**

end for

return $\{V_c, \bar{E}_c, w_c, \Psi, \Phi, v_c\}$

end function

other vertices v with weights $w(v_r, v) = f(\log \beta_{v,v})$. The function $f()$ must make the weights such that the resulting maximum weight directed tree found for this modified graph only has one edge from v_r which points to the true root. This can be achieved

by using

$$f(a) = a - \frac{1}{N}\left(\sum_{v=1}^{N} \log \beta_{v,v} - \min_{(u,v)} w(u,v)\right). \tag{A.11}$$

This function ensures that $w(v_r, v) = f(\log \beta_{v,v})$ is small enough such that even if you picked all edges from v_r the sum total weight will be less than picking the smallest edge among the vertices other than v_r. It also keeps the same relative relationships among the root weights. Calling ChowLiuEdmondsBock with v_r designated as the root will produce the MAP directed tree after removing v_r from the result.

The problem of estimating the MAP directed forest ($\bar{E} \in \mathcal{F}_N$) is also straightforward. We again use the mapping $w(u,v) = \log \beta_{u,v}$ and create a virtual root node v_r. However this time the root weights $w(v_r, v)$ are simply $\log \beta_{v,v}$. Calling ChowLiuEdmondsBock with v_r designated as the root will produce the MAP directed forest after removing v_r from the result.

■ A.4 Notes on Numerical Precision

It is important to note that one must be careful with numerical precision when using the temporal interaction model (TIM) presented in this dissertation. While performing and storing results of calculations in the log domain is good practice, there are situations in with issues which floating point accuracy cannot be avoided. For example, when dealing with large datasets (large T) the evidence weights, W, calculated in Equation 3.89 become extremely small. In addition, the ratio of the smallest evidence term to the largest can also become quite large. When reasoning over directed trees or forests these extreme values cause ill conditioned matrices $\bar{Q}(\beta \circ W)$ (see Equation 3.68) and makes accurate calculation of determinants of this matrix difficult. A determinant is required when calculating the partition function and estimating event probabilities (see Sections 3.4.2 and 3.4.3).

The datasets used in this dissertation were small enough such this issue did not arise when dealing with the partition function. However, issues arose when sampling directed trees or forests from a posterior. The evidence weights caused Wilson's RandomTreeWithRoot algorithm to take a long time to complete or in some cases loop indefinitely. The root cause of such behavior was that the stochastic graph formed using these weights had extremely large mean hitting times. Our solution to this problem was

to set a maximum number of iterations for the algorithm[2]. If the maximum number of iterations was reached we resort to an importance sampling technique using a proposal distribution based on a modified set of evidence weights. The proposal distribution used evidence weights which were rescaled so that they fit in the range $[-K, 1]$ in the log domain. Rescaling in the log domain is a nonlinear operation but maintains the relative ordering of the evidence weights, effectively broadening the proposal distribution over structure with respect to the true posterior. A similar approach was taken by [17] for undirected trees. [17] also use a library (NTL) [80] which allows them to calculate determinants to a desired precision (at the cost of speed).

[2]For the basketball data this was set to a number such that the sampler took no longer than 20 seconds to finish

Appendix B

Derivations

In this Appendix we provide details on and derivations for various equations presented in this dissertation.

■ B.1 Derivation of Equations 3.38 and 3.40

Recall that the log likelihood ratio has the following form:

$$l_{1,2} = \sum_t \log \frac{p\left(\mathcal{D}_t|\widetilde{\mathcal{D}}_t, F^1, \Theta^1\right)}{p\left(\mathcal{D}_t|\widetilde{\mathcal{D}}_t, F^2, \Theta^2\right)} \tag{B.1}$$

Taking the exception under H_1 yields:

$$\mathbb{E}_{\mathcal{D}}\left[l_{1,2}|H_1\right] = \int_{\mathcal{D}} p\left(\mathcal{D}|F^1, \Theta^1\right) \sum_t \log \frac{p\left(\mathcal{D}_t|\widetilde{\mathcal{D}}_t, F^1, \Theta^1\right)}{p\left(\mathcal{D}_t|\widetilde{\mathcal{D}}_t, F^2, \Theta^2\right)} d\mathcal{D} \tag{B.2}$$

$$= \sum_t \int_{\mathcal{D}_t, \widetilde{\mathcal{D}}_t} p\left(\mathcal{D}_t, \widetilde{\mathcal{D}}_t|F^1, \Theta^1\right) \frac{p\left(\mathcal{D}_t|\widetilde{\mathcal{D}}_t, F^1, \Theta^1\right)}{p\left(\mathcal{D}_t|\widetilde{\mathcal{D}}_t, F^2, \Theta^2\right)} d\mathcal{D}_t d\widetilde{\mathcal{D}}_t \tag{B.3}$$

$$= \sum_t \int_{\mathcal{D}_t, \widetilde{\mathcal{D}}_t} p\left(\mathcal{D}_t, \widetilde{\mathcal{D}}_t|F^1, \Theta^1\right) \frac{p\left(\mathcal{D}_t|\widetilde{\mathcal{D}}_t, F^1, \Theta^1\right)}{p\left(\mathcal{D}_t|\widetilde{\mathcal{D}}_t, F^\cap, \Theta^1\right)} \frac{p\left(\mathcal{D}_t|\widetilde{\mathcal{D}}_t, F^\cap, \Theta^1\right)}{p\left(\mathcal{D}_t|\widetilde{\mathcal{D}}_t, F^2, \Theta^2\right)} d\mathcal{D}_t d\widetilde{\mathcal{D}}_t \tag{B.4}$$

$$= \sum_t D\left(p\left(\mathcal{D}_t|\widetilde{\mathcal{D}}_t, F^1, \Theta^1\right) \middle\| p\left(\mathcal{D}_t|\widetilde{\mathcal{D}}_t, F^\cap, \Theta^1\right) \right)$$

$$+ \sum_t \int_{\mathcal{D}_t, \widetilde{\mathcal{D}}_t} p\left(\mathcal{D}_t, \widetilde{\mathcal{D}}_t|F^1, \Theta^1\right) \log \frac{p\left(\mathcal{D}_t|\widetilde{\mathcal{D}}_t, F^\cap, \Theta^1\right)}{p\left(\mathcal{D}_t|\widetilde{\mathcal{D}}_t, F^2, \Theta^2\right)} d\mathcal{D}_t d\widetilde{\mathcal{D}}_t \tag{B.5}$$

It breaks up into two terms in Equation B.5. The first term is well defined, but the second is not. Let us look at this second term and simplify notation. It can also be

broken down into two sub-terms:

$$\int_{\mathcal{D}_t,\widetilde{\mathcal{D}}_t} p_1(F^1,\widetilde{F}^1)\log\frac{p_1(F^\cap|\widetilde{F}^\cap)}{p_2(F^2|\widetilde{F}^2)} = \int_{\mathcal{D}_t,\widetilde{\mathcal{D}}_t} p_1(F^1,\widetilde{F}^1)\log p_1(F^\cap|\widetilde{F}^\cap) - \int_{\mathcal{D}_t,\widetilde{\mathcal{D}}_t} p_1(F^1,\widetilde{F}^1)\log p_2(F^2|\widetilde{F}^2) \tag{B.6}$$

where we use the notation $p_B(F^A|\widetilde{F}^A) \triangleq p\left(\mathcal{D}_t|\widetilde{\mathcal{D}}_t, F^A, \Theta^B\right)$ and drop the $d\mathcal{D}_t$ and $d\widetilde{\mathcal{D}}_t$ for space. Each sub-term can be looked at separately, starting with the first sub-term:

$$\int_{\mathcal{D}_t,\widetilde{\mathcal{D}}_t} p_1(F^1,\widetilde{F}^1)\log p_1(F^\cap|\widetilde{F}^\cap) = \int_{\mathcal{D}_t,\widetilde{\mathcal{D}}_t} p_1(F^1,\widetilde{F}^1)\log\prod_{k=1}^{|F^\cap|} p_1(F_k^\cap|\widetilde{F}_k^\cap) \tag{B.7}$$

$$= \sum_{k=1}^{|F^\cap|}\int_{\mathcal{D}_t,\widetilde{\mathcal{D}}_t} p_1(F^1,\widetilde{F}^1)\log p_1(F_k^\cap|\widetilde{F}_k^\cap) \tag{B.8}$$

$$= \sum_{k=1}^{|F^\cap|}\int_{F_k^\cap,\widetilde{F}_K^\cap} p_1(F_k^\cap,\widetilde{F}_k^\cap)\log p_1(F_k^\cap|\widetilde{F}_k^\cap) \tag{B.9}$$

$$= \sum_{k=1}^{|F^\cap|}\int_{\mathcal{D}_t,\widetilde{\mathcal{D}}_t} p_1(F^\cap,\widetilde{F}^\cap)\log p_1(F_k^\cap|\widetilde{F}_k^\cap) \tag{B.10}$$

$$= \int_{\mathcal{D}_t,\widetilde{\mathcal{D}}_t} p_1(F^\cap,\widetilde{F}^\cap)\log p_1(F^\cap|\widetilde{F}^\cap) \tag{B.11}$$

The progression from Equation B.8 to B.9 is due to the fact that all terms other than $F_k^\cap$ get marginalized and that $F_k^\cap$ is a subset of one of the factors in F^1. Equation B.9 becomes B.10 by adding back terms which will be marginalized out, this time in terms of the common factorization $F^\cap$.

The second sub-term is

$$\int_{\mathcal{D}_t,\widetilde{\mathcal{D}}_t} p_1(F^1,\widetilde{F}^1)\log p_2(F^2|\widetilde{F}^2) = -\int_{\mathcal{D}_t,\widetilde{\mathcal{D}}_t} p_1(F^1,\widetilde{F}^1)\log\prod_{j=1}^{|F^2|} p_2(F_j^2|\widetilde{F}_j^2) \tag{B.12}$$

$$= -\sum_{j=1}^{|F^2|}\int_{\mathcal{D}_t,\widetilde{\mathcal{D}}_t} p_1(F^1,\widetilde{F}^1)\log p_1(F_j^2|\widetilde{F}_j^2) \tag{B.13}$$

$$= -\sum_{j=1}^{|F^2|}\int_{F_j^2,\widetilde{F}_j^2}\int_{\mathcal{D}_t\setminus F_j^2,\widetilde{\mathcal{D}}_t\setminus F_j^2} p_1(F^1,\widetilde{F}^1)\log p_2(F_j^2|\widetilde{F}_j^2) \tag{B.14}$$

$$= -\sum_{j=1}^{|F^2|}\int_{F_j^2,\widetilde{F}_j^2}\prod_{i=1}^{|F^1|} p_1(F_i^1\cap F_j^2,\widetilde{F}_i^1\cap\widetilde{F}_j^2)\log p_2(F_j^2|\widetilde{F}_j^2) \tag{B.15}$$

$$= -\sum_{j=1}^{|F^2|}\int_{\mathcal{D}_t,\widetilde{\mathcal{D}}_t} p_1(F^\cap,\widetilde{F}^\cap)\log p_2(F_j^2|\widetilde{F}_j^2) \tag{B.16}$$

$$= -\int_{\mathcal{D}_t,\widetilde{\mathcal{D}}_t} p_1(F^\cap,\widetilde{F}^\cap)\log p_2(F^2|\widetilde{F}^2) \tag{B.17}$$

where, here, Equation B.14 is marginalizing $p_1(F^1|\widetilde{F}^1)$ over all terms but F_j^2. This marginalization yields a factorization which is formed by intersecting each factor in F^1 with F_j^2 as in Equation B.15. These intersection terms are by definition consistent with $F^\cap$ and Equation B.16 simply reintroduces all terms of $p_1(F^\cap|\widetilde{F}^\cap)$ which were being marginalized out.

Plugging Equations B.11 and B.17 into Equation B.6 and expanding notation yields

$$\int_{\mathcal{D}_t,\widetilde{\mathcal{D}}_t} p_1(F^1,\widetilde{F}^1)\log\frac{p_1(F^\cap|\widetilde{F}^\cap)}{p_2(F^2|\widetilde{F}^2)} \tag{B.18}$$

$$= \int_{\mathcal{D}_t,\widetilde{\mathcal{D}}_t} p_1(F^\cap,\widetilde{F}^\cap)\log p_1(F^\cap|\widetilde{F}^\cap) - \int_{\mathcal{D}_t,\widetilde{\mathcal{D}}_t} p_1(F^\cap,\widetilde{F}^\cap)\log p_2(F^2|\widetilde{F}^2) \tag{B.19}$$

$$= \int_{\mathcal{D}_t,\widetilde{\mathcal{D}}_t} p_1(F^\cap,\widetilde{F}^\cap)\log\frac{p_1(F^\cap|\widetilde{F}^\cap)}{p_2(F^2|\widetilde{F}^2)} \tag{B.20}$$

$$= D\left(\; p_1(F^\cap|\widetilde{F}^\cap)\;\middle\|\; p_1(F^2|\widetilde{F}^2)\;\right) \tag{B.21}$$

$$= D\left(\; p\left(\mathcal{D}_t|\widetilde{\mathcal{D}}_t,F^\cap,\Theta^1\right)\;\middle\|\; p\left(\mathcal{D}_t|\widetilde{\mathcal{D}}_t,F^2,\Theta^2\right)\;\right) \tag{B.22}$$

Plugging this into Equation B.5 we get the final result:

$$\mathbb{E}_{\mathcal{D}}\left[l_{1,2}|H_1\right] = \sum_t D\left(\ p\left(\mathcal{D}_t|\widetilde{\mathcal{D}}_t, F^1, \Theta^1\right) \Big\| p\left(\mathcal{D}_t|\widetilde{\mathcal{D}}_t, F^{\cap}, \Theta^1\right)\ \right) \tag{B.23}$$

$$+ \sum_t D\left(\ p\left(\mathcal{D}_t|\widetilde{\mathcal{D}}_t, F^{\cap}, \Theta^1\right) \Big\| p\left(\mathcal{D}_t|\widetilde{\mathcal{D}}_t, F^2, \Theta^2\right)\ \right) \tag{B.24}$$

The form of $\mathbb{E}_{\mathcal{D}}\left[l_{1,2}|H_2\right]$ can be symmetrically derived.

■ B.2 Consistent ML Estimates: Equations 3.42 through 3.45

The definition of a consistent estimator is that the estimate $\hat{\Theta}$ from data $\mathcal{D}_{1:T}$ asymptotically converges true parameter Θ as the amount of data, T, grows without bound. There are many forms of convergence. Here, we will define consistency in terms of weak convergence in distribution such that as $T \to \infty$, $p\left(\hat{\Theta}\right)$ will asymptotically approach a delta function on Θ. By this definition, if our ML estimate of Θ^1 is consistent then, given enough data under H_1, Equation 3.42 holds true. That is,

$$p\left(\mathcal{D}_t|\widetilde{\mathcal{D}}_t, F^1, \hat{\Theta}^1\right) \overset{H_1}{\to} p\left(\mathcal{D}_t|\widetilde{\mathcal{D}}_t, F^1, \Theta^1\right). \tag{B.25}$$

Next we look at what happens to $p\left(\mathcal{D}_t|\widetilde{\mathcal{D}}_t, F^1, \hat{\Theta}^2\right)$ given data under H_1. For simplicity, we first explore the case in which $r = 0$ and the quantity of interest becomes $p\left(\mathcal{D}_t|F^1, \hat{\Theta}^2\right)$. This distribution, by definition of a FactM(0) is invariant to t. An ML estimate of Θ^2 given data from H_1 takes the form:

$$\hat{\Theta}^2 = \arg\max_{\Theta} \frac{1}{T} \sum_{t=1}^{T} \log p\left(\mathcal{D}_t|F^2, \Theta\right) \tag{B.26}$$

As $T \to \infty$, using the law of large numbers the sample average converges to the expectation:

$$\hat{\Theta}^2 = \arg\max_{\Theta} \mathbb{E}_{\mathcal{D}|H_1}\left[\log p\left(\mathcal{D}_t|F^2, \Theta\right)\right] \tag{B.27}$$

$$= \arg\max_{\Theta} \int_{\mathcal{D}_t} p\left(\mathcal{D}_t|F^1, \Theta^1\right) \log p\left(\mathcal{D}_t|F^2, \Theta\right) d\mathcal{D}_t \tag{B.28}$$

$$= \arg\max_{\Theta} \int_{\mathcal{D}_t} p\left(\mathcal{D}_t|F^1, \Theta^1\right) \log \frac{p\left(\mathcal{D}_t|F^2, \Theta\right)}{p\left(\mathcal{D}_t|F^{\cap}, \Theta^1\right)} d\mathcal{D}_t \tag{B.29}$$

$$= \arg\min_{\Theta} \int_{\mathcal{D}_t} p\left(\mathcal{D}_t|F^1, \Theta^1\right) \log \frac{p\left(\mathcal{D}_t|F^{\cap}, \Theta^1\right)}{p\left(\mathcal{D}_t|F^2, \Theta\right)} d\mathcal{D}_t \tag{B.30}$$

$$= \arg\min_{\Theta} D\left(\ p\left(\mathcal{D}_t|F^{\cap}, \Theta^1\right) \Big\| p\left(\mathcal{D}_t|F^2, \Theta\right)\ \right) \tag{B.31}$$

where B.29 introduces a term which is constant with respect to Θ and the last line uses our derivation of Equation B.22 above. Note that $p\left(\mathcal{D}_t|F^2,\Theta\right)$ is a more expressive model than $p\left(\mathcal{D}_t|F^{\cap},\Theta^1\right)$ since the factors in $F^{\cap}$ are subsets of the factors in F^2 by definition. Thus,

$$\arg\min_{\Theta} D\left(\ p\left(\mathcal{D}_t|F^{\cap},\Theta^1\right)\ \middle\|\ p\left(\mathcal{D}_t|F^2,\Theta\right)\ \right)=0 \tag{B.32}$$

and

$$p\left(\mathcal{D}_t|F^2,\hat{\Theta}^2\right)\overset{H_1}{\rightarrow}p\left(\mathcal{D}_t|F^{\cap},\Theta^1\right) \tag{B.33}$$

That is, the ML estimate of Θ^2 is the one that produces $p\left(\mathcal{D}_t|F^2,\hat{\Theta}^2\right)$ most consistent with the true hypothesis under the common factorization.

The $r=0$ case is used in all the examples and experiments presented in this dissertation. Intuitively, a similar relationship should hold true for $r>0$. However, for $r>0$ one cannot use the law of large numbers to turn the sample average into an expectation because one does not obtain independent samples over time. That is, for $r>0$:

$$\hat{\Theta}^2=\arg\max_{\Theta}\frac{1}{T}\sum_{t=1}^{T}\log p\left(\mathcal{D}_t|\widetilde{\mathcal{D}}_t,F^2,\Theta\right) \tag{B.34}$$

where each $\mathcal{D}_t$ is no longer independent of all other times. If one could prove that as $T\rightarrow\infty$,

$$\frac{1}{T}\sum_{t=1}^{T}\log p\left(\mathcal{D}_t|\widetilde{\mathcal{D}}_t,F^2,\Theta\right)\rightarrow\mathbb{E}_{\mathcal{D}|H1}\left[\log p\left(\mathcal{D}_t|\widetilde{\mathcal{D}}_t,F^2,\Theta\right)\right] \tag{B.35}$$

then one could follow the same form of analysis above to show:

$$p\left(\mathcal{D}_t|\widetilde{\mathcal{D}}_t,F^2,\hat{\Theta}^2\right)\overset{H_1}{\rightarrow}p\left(\mathcal{D}_t|\widetilde{\mathcal{D}}_t,F^{\cap},\Theta^1\right) \tag{B.36}$$

Equation B.35 holds true when our static dependence model is ergodic and stationary. If stationary, all joint distributions are invariant to t. If ergodic, sample averages over time converge to ensemble averages.

■ B.3 Derivation of Expectations of Additive Functions over Structure

Equation 3.105: For $\bar{E} \in \mathcal{A}_N$ or $\bar{E} \in \mathcal{P}_N^K$ the expectation over additive function f takes the form

$$\mathbb{E}\left[f(\bar{E})\right] = \sum_{\bar{E}} p_0\left(\bar{E}\right) f(\bar{E}) \tag{B.37}$$

$$= \sum_{\bar{E}} \frac{1}{Z(\beta)} \prod_{i=1} \beta_{\mathbf{pa}(i),i} \sum_{j=1}^{N} f_{\mathbf{pa}(j),j} \tag{B.38}$$

$$= \frac{1}{Z(\beta)} \sum_{\mathbf{S}_1} \cdots \sum_{\mathbf{S}_N} \prod_{i=1} \beta_{\mathbf{S}_i,i} \sum_{j=1}^{N} f_{\mathbf{S}_j,j} \tag{B.39}$$

$$= \frac{1}{Z(\beta)} \sum_{\mathbf{S}_1} \cdots \sum_{\mathbf{S}_N} \sum_{j=1}^{N} f_{\mathbf{S}_j,j} \prod_{i=1} \beta_{\mathbf{S}_i,i} \tag{B.40}$$

$$= \frac{1}{Z(\beta)} \sum_{j=1}^{N} \sum_{\mathbf{S}_1} \cdots \sum_{\mathbf{S}_N} f_{\mathbf{S}_j,j} \prod_{i=1} \beta_{\mathbf{S}_i,i} \tag{B.41}$$

$$= \frac{1}{Z(\beta)} \sum_{j=1}^{N} \sum_{\mathbf{S}_j} f_{\mathbf{S}_j,j} \beta_{\mathbf{S}_j,j} \sum_{\mathbf{S}_k \ \forall \ k \in V \setminus j} \prod_{i \in V \setminus j} \beta_{\mathbf{S}_i,i} \tag{B.42}$$

$$= \frac{1}{Z(\beta)} \sum_{j=1}^{N} \left(\sum_{\mathbf{S}_j} f_{\mathbf{S}_j,j} \beta_{\mathbf{S}_j,j} \right) \left(\prod_{i \in V \setminus j} \sum_{\mathbf{S}_i} \beta_{\mathbf{S}_i,i} \right) \tag{B.43}$$

$$= \frac{1}{Z(\beta)} \sum_{j=1}^{N} \left(\gamma_j(\beta \circ f)\right) \left(\prod_{i \in V \setminus j} \gamma_i(\beta) \right) \tag{B.44}$$

$$= \frac{1}{Z(\beta)} \sum_{j=1}^{N} \left(\gamma_j(\beta \circ f)\right) \left(\frac{Z(\beta)}{\gamma_j(\beta)} \right) \tag{B.45}$$

$$= \sum_{j=1}^{N} \frac{\gamma_j(\beta \circ f)}{\gamma_j(\beta)} \tag{B.46}$$

Equation 3.106: If $\bar{E} \in \mathcal{T}_N$ then

$$\begin{aligned} \mathbb{E}\left[f(\bar{E})\right] &= \mathbb{E}_r\left[\mathbb{E}\left[f(\bar{E}) | \text{the root is } r\right]\right] \\ &= \sum_{r-1}^{N} p\left(r\right) \mathbb{E}\left[f(\bar{E})|r\right] \\ &= \sum_{r=1}^{N} \frac{Z_r(\beta)}{Z(\beta)} \text{tr}\left(M_{r,r}\left(\bar{Q}(\beta \circ f)\right) M_{r,r}\left(\bar{Q}(\beta)\right)^{-1} \right) \end{aligned} \tag{B.47}$$

The last line use the form for the probability of a root and the expectation over an additive function given a root follows the form shown in [62] substituting in the directed tree partition function in place of the undirected version.

■ B.4 Derivation of Equations 4.20 and 4.21

The likelihood ratio comparing two hypothesized state sequences with $r = 0$ takes the form:

$$\hat{l}_{1,2} = \log \frac{p\left(\mathcal{D}|a_{1:T}, \hat{\mathbf{F}}, \hat{\mathbf{\Theta}}\right)}{p\left(\mathcal{D}|b_{1:T}, \hat{\mathbf{F}}, \hat{\mathbf{\Theta}}\right)} \tag{B.48}$$

$$= \sum_{t=1}^{T} \log \frac{p\left(\mathcal{D}_t|\hat{F}^{a_t}, \hat{\Theta}^{a_t}\right)}{p\left(\mathcal{D}_t|\hat{F}^{b_t}, \hat{\Theta}^{b_t}\right)} \tag{B.49}$$

$$= \sum_{t\in\mathbf{d}} \log \frac{p\left(\mathcal{D}_t|\hat{F}^{a_t}, \hat{\Theta}^{a_t}\right)}{p\left(\mathcal{D}_t|\hat{F}^{b_t}, \hat{\Theta}^{b_t}\right)} \tag{B.50}$$

$$= \sum_{t\in\mathbf{d}} \log \frac{p\left(\mathcal{D}_t|\hat{F}^{a_t}, \hat{\Theta}^{a_t}\right)}{p\left(\mathcal{D}_t|\hat{F}^{\cap_t}, \hat{\Theta}^{a_t}\right)} \frac{p\left(\mathcal{D}_t|\hat{F}^{\cap_t}, \hat{\Theta}^{a_t}\right)}{p\left(\mathcal{D}_t|\hat{F}^{b_t}, \hat{\Theta}^{b_t}\right)} \tag{B.51}$$

$$= \sum_{t\in\mathbf{d}} \log \frac{p\left(\mathcal{D}_t|\hat{F}^{a_t}, \hat{\Theta}^{a_t}\right)}{p\left(\mathcal{D}_t|\hat{F}^{\cap_t}, \hat{\Theta}^{b_t}\right)} \frac{p\left(\mathcal{D}_t|\hat{F}^{\cap_t}, \hat{\Theta}^{b_t}\right)}{p\left(\mathcal{D}_t|\hat{F}^{b_t}, \hat{\Theta}^{b_t}\right)} \tag{B.52}$$

where $\mathbf{d} = \{t|a_t \neq b_t\}$ is the set of all time points in which the hypothesized state sequences differ. Equations 4.20 and 4.21 are obtained by taking the expectation of $\hat{l}_{1,2}$ over data drawn from each hypothesized state sequence and using the same form of derivation shown in Section B.1 with $r = 0$. That is, for each t we simply map a_t to H_1 and b_t to H_2.

Bibliography

[1] C. Andrieu, N. de Freitas, A. Doucet, and M. I. Jordan. An introduction to MCMC for machine learning. In *Machine Learning*, volume 50, pages 5–43, 2003. 102, 104

[2] D. Ayers and M. Shah. Monitoring human behavior from video taken in an office environment. *Image and Vision Computing*, 19(1):833–846, 2001. 137

[3] O. Barndor-Nielsen. *Information and Exponential Families*. John Wiley, 1978. 37

[4] L.E. Baum and J.A. Egon. An inequality with applications to statistical estimation for probabilistic functions of a Markov process and to a model for ecology. In *Bull. Amer. Meteorol. Soc.*, volume 73, pages 360–363, 1967. 90, 91

[5] D. P. Bertsekas and J.N. Tsitsiklis. *Introduction to Probability*. Athena Scientific, 2002. 27

[6] P. Besson, G. Monaci, P. Vandergheynst, and M. Kunt. Experimental framework for speaker detection on the CUAVE database. In *Technical Report 2006-003, EPFL, Lausanne, Switzerland*, 2006. 112, 121, 122

[7] J. A. Bilmes. Dynamic Bayesian multinets. In *UAI*, 2000. 108, 109

[8] A. L. Blum and P. Langely. Selection of relevant features and examples in machine learning. In *Artificial Intelligence*, 1997. 20

[9] A.F. Bobick and J.W. Davis. The recognition of human movement using temporal templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(3):257–267, March 2001. 137

[10] F. Bock. An algorithm to construct a minimum spanning tree in a directed network. Technical report, Developments in Operations Research, Gordon and Breach, NY, 1971. 80, 170

[11] C. Boutilier, N. Friedman, M. Goldszmidt, and D. Koller. Context-specific independence in Bayesian networks. In *UAI*, pages 115–123, 1996. 108

[12] M. Brand. Learning concise models of human activity from ambient video. Technical Report 97-25, Mitsubishi Electric Research Labs, 1997. 137

[13] J. S. Bridle and M. D. Brown. An experimental automatic word-recognition system. Technical Report 1003, Joint Speech Research Unit, Ruislip, England., 1974. 116

[14] H. Buxton. Generative models for learning and understanding dynamic scene activity. In *1st International Workshop on Generative-Model-Based Vision*, 2002. 137

[15] C. Castel, L. Chaudron, and C. Tessier. What is going on? A high level interpretation of sequences of images. In *European Conference on Computer Vision*, pages 13–27, 1996. 137

[16] A. Cayley. A theorem on trees. *Quart. J. Math.*, 23:376–378, 1889. 69

[17] J. Cerquides and R.L. de Màntaras. TAN classifiers based on decomposable distributions. In *Machine Learning*, volume 59, pages 1–32, 2005. 174

[18] Y. J. Chu and T. H. Liu. On the shortest arborescence of a directed graph. *Science Sinica*, 14:1396–1400, 1965. 80, 170

[19] C. J. Colbourn, W.J. Myrvold, and E. Neufeld. Two algorithms for unranking arborescences. In *Journal of Algorithms*, pages 268–281, 1996. 164, 166

[20] R. Collins, A. Lipton, and T. Kanade. A system for video surveillance and monitoring. In *American Nuclear Society Eight International Topical Meeting on Robotic and Remote Systems*, 1999. 137

[21] T.M. Cover and J.A. Thomas. *Elements of Information Theory*. John Wiley and Sons, Inc., 1991. 27, 43

[22] N. G. de Bruijn. *Asymptotic Methods in Analysis*. Dover Publications, 1981. 56

[23] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. In *Journal of the Royal Statistical Society, Series B*, volume 39, pages 1–38, 1977. 91

[24] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. John Wiley & Sons, 1973. 100

[25] J. Edmonds. Optimum branchings. *J. Research of the National Bureau of Standards*, 71B:233–240, 1967. 80, 170

[26] P. Fearnhead. Exact and efficient bayesian inference for multiple changepoint problems. In *Statistics and Computing*, volume 16, 2006. 158

[27] P. Fearnhead and P. Clifford. On-line inference for hidden Markov models via particle filters. In *Journal of the Royal Statistical Society, Series B*, volume 65, pages 887–899, 2003. 157

[28] P. Fearnhead and Z. Liu. Online inference for multiple changepoint problems. In *Journal of the Royal Statistical Society, Series B*, volume 69, 2007. 158

[29] J.W. Fisher, III and J.C. Principie. A methodology for information theoretic feature extraction. In *Proceedings of the IEEE International Joint Conference on Neural Networks*, pages 772–778, 1998. 20

[30] J.W. Fisher, III, T.Darrell, W.T. Freeman, and P.A. Viola. Learning joint statistical models for audio-visual fusion and segregation. In *NIPS*, pages 772–778, 2000. 20, 22, 111, 115

[31] G.D. Forney. The viterbi algorithm. In *IEEE*, pages 268–278, 1973. 92

[32] N. Friedman and M. Goldszmidt. Learning Bayesian networks with local structure. In *Learning in Graphical Models*, pages 252–262. MIT Press, 1996. 108

[33] N. Friedman and D. Koller. Being Bayesian about network structure. a bayesian approach to structure discovery in Bayesian networks. In *Machine Learning*, volume 50, pages 95–125, 2003. 24, 48, 70

[34] J. Garofolo, C. Laprum, M. Michel, V. Stanford, and E. Tabassi. The NIST meeting room pilot corpus. In *Language Resource and Evaluation Conference*, 2004. 114

[35] D. Geiger and D. Heckerman. Knowledge representation and inference in similarity networks and Bayesian multinets. In *Artificial Integlligence*, volume 82, pages 45–74, 1996. 107

[36] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian Data Analysis*. Chapman & Hall, 2004. 102, 104

[37] S. Geman and D. Geman. Gibbs distributions, and the Bayesian restoration of images. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 6(6), pages 721–741, 1984. 105

[38] P. Good. *Permutation Tests: A Practical Guide to Resampling Methods for Testing Hypotheses*. Springer-Verlag, 1994. 62

[39] W.E.L. Grimson, L. Lee, R. Romano, and C. Stauffer. Using adaptive tracking to classify and monitor activities in a site. In *Computer Vision and Pattern Recognition*, 1998. 137

[40] M. Gurban and J. Thiran. Multimodal speaker localization in a probabilistic framework. In *Proceedings of EUSIPCO*, 2006. 112, 115, 121, 122

[41] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. In *Journal of Machine Learning Research*, 2003. 20

[42] P. Hall. On the bootstrap and confidence intervals. *Annals of Statistcs*, 14(4):1431–1452, 1986. 62

[43] R.W. Hamming. Error detecting and error correcting codes. In *Bell System Technical Journal*, volume 26, 1950. 147

[44] D. Heckerman. *Probabilistic Similarity Networks*. MIT Press, Cambridge, Massachusetts, 1991. 107

[45] F. Heider and M. Simmel. An experimental study of apparent behavior. In *American Journal of Psychology*, volume 57, pages 243–256, 1944. 21

[46] J. Hershey and J. Movellan. Audio-vision: Using audio-visual synchrony to locate sounds. In *Neural Information Processing Systems*, pages 813–819, 1999. 22, 111, 115

[47] A.T. Ihler, J.W. Fisher, and A.S. Willsky. Nonparameteric hypothesis tests for statistical dependency. In *Trans. on signal processing, special issue on machine learning*, 2004. 59, 62, 63

[48] S.S. Intille and A.F. Bobick. A framework for recognizing multi-agent action from visual evidence. In *AAAI*, pages 518–525, 1999. 22

[49] Y.A. Ivanov, A.F. Bobick, C. Stauffer, and W.E.L. Grimson. Visual surveillance of interactions. In *International Workshop on Visual Surveillance*, pages 82–89, 1999. 22

[50] I.T. Jolliffe. *Principal Component Analysis*. Springer, 2002. 116

[51] G. Kirchhoff. Ueber die auflösung der gleichungen, auf welche man bei der untersuchung der linearen vertheilung galvanischer ströme geführt wird. In *Annalen der Physik und Chemie*, volume 72, pages 497–508, 1847. 73

[52] S. Kirshner, P. Smyth, and A. Robertson. Conditional chow-liu tree structures for modeling discrete-valued vector time series. In *Technical Report UCI-ICS*, 2004. 109

[53] R. Kohavi and G.H. John. Wrappers for feature subset selection. In *Artificial Intelligence*, 1997. 20

[54] M. Koivisto, K. Sood, and M. Chickering. Exact Bayesian structure discovery in Bayesian networks. In *Journal of Machine Learning Research*, volume 5, 2004. 24, 48

[55] D. Koller and M. Sahami. Toward optimal feature selection. In *13th International Conference on Machine Learning*, 1996. 20

[56] T. Koo, A. Globerson, X. Carreras, and M. Collins. Structured prediction models via the matrix-tree theorem,. In *EMNLP*, 2007. 73, 74

[57] B. Korte and J. Vygen. *Combinatorial Optimization : Theory and Algorithms, Third Edition*. Springer, 2000. 170

[58] F.R. Kschischang, B.J. Frey, and H-A Loeliger. Factor graphs and the sum-product algorithm. 47(2):498–519, 2001. 30

[59] S. Kullback. *Information Theory and Statistics*. 1968. 43

[60] B-H. Juang L. R. Rabiner. *Fundamentals of Speech Recognition*. PTR Prentice Hall, 1993. 111

[61] R. McDonald, F. Pereira, K. Ribarov, and J. Hajič. Non-projective dependency parsing using spanning tree algorithms. In *HLT/EMNLP*, 2005. 170

[62] M. Meila and T. Jaakkola. Tractable Bayesian learning of tree belief networks. In *Statistical Computing*, pages 77–92, 2006. 24, 70, 73, 76, 181

[63] P. Mermelstein. Distance measures for speech recognition, psychological and instrumental. In *Pattern Recognition and Artificial Intelligence*, 1976. 116

[64] B. Milch, B. Marthi, D. Sontag, S. Russell, D.L. Ong, and A. Kolobov. Approximate inference for infinite contingent Bayesian networks. In *AIStats*, 2005. 108

[65] T. Minka. Expectation-maximization as lower bound maximization, 1998. 91

[66] R. J. Morris and David C. Hogg. Statistical models of object interaction. *International Journal of Computer Vision*, 37(2):209–215, 2000. 22, 137

[67] K.P. Murphy. *Dynamic Bayesian Networks: Representation, Inference and Learning*. PhD thesis, UC Berkeley, Computer Science Division, 2002. 157

[68] H.J. Nock, G. Iyengar, and C. Neti. Speaker localisation using audio-visual synchrony: An empirical study. In *Proc. Intl. Conf. on Image and Video Retrieval*, 2003. 22, 23, 111, 112, 115, 121, 122

[69] N. M. Oliver, B. Rosario, and A. Pentland. A Bayesian computer vision system for modeling human interactions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):831–843, 2000. 22

[70] A. Papoulis. *Probability, Random Variables and Stochastic Processes*. McGraw-Hill, third edition, 1991. 27, 139

[71] V. Parameswaran and R. Chellappa. Human action-recognition using mutual invariants. *Computer Vision and Image Understanding*, 98(2):294–324, May 2005. 137

[72] E.K. Patterson, S. Gurbuz, Z. Tufekci, and J.N. Gowdy. CUAVE: A new audio-visual database for multimodal human-computer interface research. Technical report, Department of ECE, Clemson University, 2001. 22, 23, 111, 112

[73] J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, 1988. 27, 29

[74] K. Pearson. On lines and planes of closest fit to systems of points, 1901. 116

[75] J. Pers, M. Bon, and G. Vuckovic. CVBASE 06 dataset. In *http://vision.fe.uni-lj.si/cvbase06/dataset.html*, 2006. 149

[76] L.R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In *Proc. IEEE*, volume 77 No. 2, pages 257–286, 1989. 91, 92

[77] J. P. Romano. Bootstrap and randomization tests of some nonparametric hypotheses. *Annals of Statistcs*, 17(1):141–159, 1989. 62

[78] G-C Rota. The number of partitions of a set. *American Mathematical Monthly*, 75(5):498–504, 1964. 56

[79] R. Shachter. Bayes-ball: The rational pastime (for determining irrelevance and requisite information in belief networks and influence diagrams. pages 480–487, 1998. 32

[80] V. Shoup. NTL: A library for doing number theory, 2003. 174

[81] M. R. Siracusa, , and J. W. Fisher. Dynamic dependency tests: Analysis and applications to multi-modal data association. In *AIStats*, 2007. 89

[82] M. R. Siracusa, , and J. W. Fisher. Tractable bayesian inference of time-series dependence structure. In *AIStats*, 2009. 89

[83] M. R. Siracusa, K. Tieu, A. T. Ihler, J. W. Fisher, and A. S. Willsky. Estimating dependency and significance for high-dimensional data. In *Acoustics, Speech, and Signal Processing*, volume 5, pages 1085–1088, 2005. 62

[84] M. Slaney and M. Covell. Facesync: A linear operator for measuring synchronization of video facial images and audio tracks. In *Neural Information Processing Systems*, 2000. 20, 22, 111, 115

[85] C. Stauffer and W.E.L. Grimson. Learning patterns of activity using real-time tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):747–757, August 2000. 137

[86] R.E. Tarjan. Finding optimum branchings. *Networks*, 1977. 170

[87] Y.W. Teh, M.I. Jordan, M.J. Beal, and D.M. Blei. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101(476), 2006. 89

[88] K. Tieu. *Statistical Dependence Estimation for Object Interaction and Matching*. PhD thesis, Massachusetts Institute of Technology, 2006. 22, 137, 142

[89] M. Turk and A. Pentland. Eigenfaces for recognition. In *Journal of Cognitive Neuroscience*, 1991. 111

[90] W.T. Tutte. *Graph Theory*. Addison-Wesley Publishing Company, 1984. 73

[91] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. Conf. on Computer Vision and Pattern Recognition*, 2001". 111, 112

[92] A.J. Viterbi. Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. In *IEEE Transactions on Information Theory*, pages 260–269, 1967. 92

[93] D. B. Wilson. Generating random spanning trees more quickly than the cover time. In *Theory of Computing*, pages 296–303, 1996. 165, 166

[94] X. Xuan and K. Murphy. Modeling changing dependency structure in multivariate time series. In *ICML*, 2007. 89, 109

[95] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld. Face recognition: A literature survey. *ACM Comput. Surv.*, 35(4):399–458, 2003. 111